


ДОДАТОК Б
Технічне завдання

Міністерство освіти та науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри БМІОЕС

 к.т.н., доц. Коваль Л.Г.


(прізвище та ініціали)

« 18 » 09 2023 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

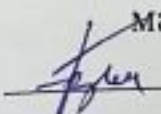
«Телемедична система передачі та попередньої обробки медичних даних»

 Науковий керівник

к.т.н., доц. каф. ОТ

Кожем'яко А.В.

виконав:


 магістрант 2 курсу,

Бондар І.Д

Вінницький національний технічний університет
Факультет інформаційних електронних систем
Кафедра біомедичної інженерії та оптико-електронних систем
Рівень вищої освіти II-й (магістерський)
Галузь знань – 15- автоматизація та приладобудування
Спеціальність – 152 - метрологія та інформаційно-вимірювальна техніка
Освітньо-професійна програма – лазерна техніка та оптоінформатика

ЗАТВЕРДЖУЮ

Завідувач кафедри БМІОЕС

 **Л.Г. Коваль**

8.09. 2023 року

ЗАВДАННЯ


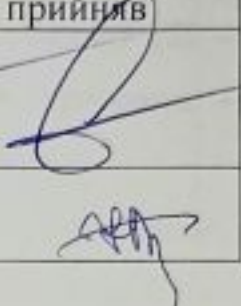

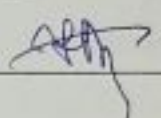
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Бондару Івану Дмитровичу

(прізвище, ім'я, по батькові)

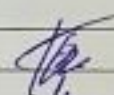
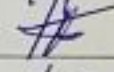
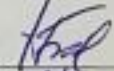
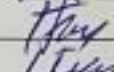

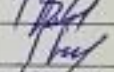
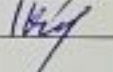
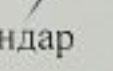
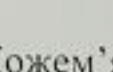
1. Тема роботи: Телемедична система передачі та попередньої обробки медичних даних
керівник роботи Кожем'яко Андрій Вікторович, к.т.н., доцент
затверджені наказом вищого навчального закладу від 18.09.2023 р., №247
2. Строк подання студентом магістерської дипломної роботи: 07.12.2023
3. Вихідні дані до роботи:
 - 3.1 В сфері телемедицини алгоритм Хаффмана виконує ключову роль, забезпечуючи ефективне стиснення та розпакування медичних даних. Це виявляється критичним аспектом для забезпечення надійної та оперативної комунікації в онлайн-медичинському середовищі. Його застосування дозволяє оптимізувати передачу інформації, зменшуючи обсяг даних, що передаються, і, таким чином, забезпечуючи високу швидкість обміну даними між медичними працівниками та пацієнтами.
4. Зміст текстової частини: Вступ : Вступ 1. Огляд стану проблеми. Телемедицина, перспективні сучасні технології. 2. Дослідження стиснення даних за допомогою алгоритму Хаффмана 3. Опис реалізації програмного застосунку. 4. Економічна частина. Висновки. Список використаних джерел. Додатки.
5. Перелік ілюстративного матеріалу: 1. Схема архітектури взаємодії клієнта з програмою. 2. Блок-схема адаптивного алгоритму Хаффмана

6. Консультанти розділів роботи

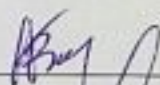
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Спеціальна частина, розділи 1-3	Кожем'яко А.В., к.т.н., доцент каф. БМІОЕС		
Економічна частина	Буреннікова Н.В., д.е.н., проф. каф. ЕПВМ		

7. Дата видачі завдання « 8 » 09 09 2023 р

КАЛЕНДАРНИЙ ПЛАН


№ з/п	Назва етапів МКР	Строк виконання етапів роботи	Примітка
1	Постановка задачі дослідження	18.09.2023	
2	Аналіз методів і систем керування розумними будинками	09.10.2023	
3	Розробка методу енергозабезпечення розумного будинка	09.11.2023	
4	Експериментальні дослідження системи	15.11.2023	
5	Економічний розділ	28.11.2023	
6	Попередній захист МКР	29.11.2023	
7	Нормоконтроль МКР	06.12.2023	
8	Рецензування МКР	07.12.2023	
8	Остаточний Захист МКР	14.12.2023	

Студент



І.Д.Бондар

Керівник роботи



А.В. Кожем'яко

АНОТАЦІЯ

Бондар І.Д. Телемедична система передачі та попередньої обробки медичних даних. Магістерська кваліфікаційна робота зі спеціальності 152 - Метрологія та інформаційно-вимірвальна техніка, освітня програма - .Лазерна техніка та оптоінформатика. Вінниця: ВНТУ, 2023. 99 с.

Метою роботи є концепція телемедицини. Телемедицина — це сучасне, зростаюче поняття як в розвинених, так і в країнах, що розвиваються. Донедавна основними носіями діагностичної інформації були рентгенівська плівка та термопапір. Але на сьогоднішній день інтенсивно використовуються цифрові можливості для передачі та обміну медичними зображеннями в комп'ютерних мережах. Завдяки цифровому обміну діагностичними зображеннями відбувається розквіт телемедицини, де застосовуються економічно виправдані, технічно та організаційно збудовані оптимальні сеанси дистанційного консультування. Покращення існуючих та розробка нових телемедичних інформаційних систем визначаються як ключові фактори у підвищенні якості та доступності кваліфікованої медичної допомоги для населення.

ABSTRACT

Bondar I.D Telemedical System for Transmission and Preliminary Processing of Medical Data. Master's Qualification Work in the field of 152 - Metrology and Information-Measuring Technology, Educational Program - Laser Technology and Optoinformatics. Vinnytsia: Vinnytsia National Technical University, 2023. 99 p.

The aim of the work is the concept of telemedicine. Telemedicine is a modern, growing concept in both developed and developing countries. Until recently, the primary carriers of diagnostic information were X-ray film and thermal paper. However, digital capabilities are now actively utilized for the transmission and exchange of medical images in computer networks. The flourishing of telemedicine is facilitated by digital exchange of diagnostic images, employing economically justified, technically, and organizationally optimized sessions of remote consultation. The improvement of existing and the development of new telemedical information systems are identified as key factors in enhancing the quality and accessibility of qualified medical assistance to the population.

ЗМІСТ

ВСТУП.....	6
1 ОГЛЯД СТАНУ ПРОБЛЕМИ. ТЕЛЕМЕДИЦИНА : ПЕРСПЕКТИВНІ СУЧАСНІ ТЕХНОЛОГІЇ.....	8
1.1 Поняття телемедицини.....	8
1.2 Складові частин телемедицини	15
1.2.1 Телемедичне консультування.....	15
1.2.2 Біотелеметрія.....	16
1.2.3 Домашня(персональна) телемедицина	17
1.2.4 Телескринінг.....	18
1.2.5 Телеприсутність	19
1.2.6 Дистанційне навчання	20
1.3 Технічний аналіз багатосервісної волоконно-оптичної мережі для телемедицини.	21
1.4 Архівація даних	25
РОЗДІЛ2. ОПИС ОСНОВНИХ ПОНЯТЬ У СТИСНЕННІ ДАНИХ З МЕТОЮ ОПТИМІЗАЦІЇ ПРОЦЕСУ ПЕРЕДАЧІ ДАНИХ.....	28
2.1. Ентропія	28
2.2. Принцип роботи Хаффмана.....	38
2.3 Кодування Хаффманом наборів символів.....	49
2.4. Застосування алгоритму Хаффмана.....	50
2.5 Оцінка стиснення даних	55
3 ОПИС РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАСТОСУНКУ	62
3.1 Засоби втілення програмного алгоритму стиснювання даних.....	62
3.2 Розробка програмної частини.....	65
3.3. Опис інтерфейсу програми	70
3.4Тестування програми.....	74
4 ЕКОНОМІЧНА ЧАСТИНА.....	79
4.1 Оцінювання наукового ефекту	79
4.2 Розрахунок витрат на здійснення науково-дослідної роботи.....	82
4.2.1 Витрати на оплату праці.....	83
4.2.2 Відрахування на соціальні заходи.....	85
4.2.3 Сировина та матеріали	86
4.2.4 Розрахунок витрат на комплектуючі	87

4.2.5 Спецустаткування для наукових (експериментальних) робіт.....	88
4.2.6 Програмне забезпечення для наукових (експериментальних) робіт....	89
4.2.7 Амортизація обладнання, програмних засобів та приміщень.....	90
4.2.8 Паливо та енергія для науково-виробничих цілей	91
4.2.9 Службові відрядження.....	92
4.2.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації	92
4.2.11 Інші витрати.....	93
4.2.12 Накладні (загальновиробничі) витрати	93
4.3 Оцінювання важливості та наукової значимості науково-дослідної роботи	94
4.4 Висновок до розділу 4	96
ВИСНОВКИ.....	97
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	99
ДОДАТОК А	104
ДОДАТОК Б.....	105
ДОДАТОК В	108
ДОДАТОК Г	109
ДОДАТОК Д.....	113
ДОДАТОК Е	114

ВСТУП

Актуальність роботи. У наш час телемедицина, як передова галузь медичної науки, стала необхідною складовою сучасної системи охорони здоров'я. За допомогою стрімкого розвитку інформаційних технологій, телемедицина вкрай розширила можливості надання медичних послуг та забезпечення доступу до них у будь-якому куточку світу. Це наукове напрямком охоплює різноманітні аспекти, розпочинаючи від телемедичних консультацій і завершуючи біотелеметрією та особистою телемедициною.

Особливу увагу ми приділяємо технічному аналізу багатосервісної волоконно-оптичної мережі для телемедицини та аспектам архівації даних, розглядаючи їхню важливість для ефективного зберігання та обміну медичною інформацією.

Також необхідно приділяти увагу принципам теорії інформації, зосереджуючись на ентропії та алгоритмах стиснення даних, зокрема на принципах роботи Хаффмана. Проводячи аналіз кодування Хаффманом наборів символів та застосування цього алгоритму, визначено ефективність стиснення даних та оцінюємо його вплив на обробку та передачу інформації у сфері телемедицини.

Наша робота націлена на розуміння та оптимізацію використання технологій у телемедицині, а також на висвітлення теоретичних аспектів інформаційних процесів, які відіграють ключову роль у сучасній медичній практиці.

Метою магістерської кваліфікаційної роботи є розширення функціональних можливостей телемедичної системи шляхом організації ефективного обміну медичними зображеннями у телемедичній мережі шляхом застосування методів стиснення даних за допомогою методу Хаффмана.

Для досягнення мети потрібно вирішити такі **задачі**:

- Розглянути поняття телемедицини;
- Провести аналіз складових частин телемедицини.

- Виконати технічний аналіз багатосервісної волоконно-оптичної мережі для телемедицини.
- Розробити метод архівації даних для задач телемедицини
- Розробити програмний засіб кодування даних за принципом роботи Хаффмана.
- Оцінити результати стиснення даних, які були отриманні.

Об’єкт дослідження - процес архівації даних, ефективного стиснювання та безпечного обміну медичною інформацією.

Предметом дослідження є методи та засоби телемедичного консультування, біотелеметрії, дистанційного навчання та інші компоненти телемедицини з метою вдосконалення надання медичних послуг та забезпечення ефективного обміну інформацією між медичними закладами та пацієнтами.

Наукова новизна. Набув подальшого розвитку метод організації сучасної телемедичної мережі, яка може застосуватись для обміну медичною інформацією з різних типів медичних засобів, який відрізняється від існуючих застосуванням методів стиснення біомедичної інформації на основі методу Хоффмана, що підвищило захищеність даних і надійність функціонування системи застосуванням апаратних засобів передачі та кодування.

За тематикою магістерської кваліфікаційної роботи були опубліковані тези на Науково-технічна конференції ВНТУ 2022 р.

1 ОГЛЯД СТАНУ ПРОБЛЕМИ. ТЕЛЕМЕДИЦИНА : ПЕРСПЕКТИВНІ СУЧАСНІ ТЕХНОЛОГІЇ

1.1 Поняття телемедицини

Телемедицина - це послуга, пов'язана з охороною здоров'я, за допомогою телекомунікаційних та електронних інформаційних технологій. Вона охоплює весь комплекс послуг, спрямованих на можливість спілкування пацієнтів з їхніми лікарями або медичними працівниками. Телемедицина має широкий спектр застосувань, включаючи онлайн консультації з пацієнтами, дистанційний моніторинг, телемедичну сестринську допомогу, а також дистанційну фізичну та психіатричну реабілітацію. Вона дозволяє приймати кращі рішення щодо охорони здоров'я, підвищує якість та ефективність надання невідкладної медичної допомоги, зменшує час, потрібний для постановки діагнозу і економить витрати як для лікарів, так і для пацієнтів, оптимізуючи клінічні процедури та зменшуючи витрати на подорожі до лікарень.

Телемедицина збільшує доступ до високоякісних медичних послуг. Тепер пацієнти можуть отримувати більш індивідуальні медичні послуги. Вони можуть також легко зустрічатися з кращими медичними постачальниками, використовуючи відео-програмне забезпечення для консультацій здалеку, а лікарі мають покращені інструменти для мережевого спілкування, зберігання даних, управління звітами і використання спеціалізованих навичок один одного. Це підвищує якість медичної практики, дозволяючи лікарям проводити менше часу в сільських районах і надавати більше допомоги пацієнтам. Телемедицина також дозволяє спеціалістам з приватних медичних закладів працювати і покращує досвід пацієнтів. Пацієнтам більше не доводиться стояти в довгих чергах, а лікарі можуть зручно та ефективно отримувати доступ до інформації про пацієнтів за допомогою електронних файлів та скорочувати загальний час очікування.

Крім того, дистанційні призначення дозволяють лікарям відводити менше часу на кожного пацієнта, що дозволяє їм надавати послуги більшій кількості пацієнтів.

Зростаючі витрати на охорону здоров'я та потреба у кращому лікуванні підштовхують більше лікарень досліджувати переваги телемедицини. Вони бажають покращити зв'язок між лікарями та пацієнтами, які знаходяться віддалено, і краще використовувати медичні заклади. Тут телемедицина також сприяє поліпшенню зв'язку, що призводить до зменшення повторних госпіталізацій та повного дотримання пацієнтами рецептів на лікування. Перевага збільшеного зв'язку, яку надає телемедицина, поширюється також на комунікацію між лікарями. Лікарі можуть використовувати телемедицину для створення мереж підтримки, обміну своїми навичками та надання кращих медичних послуг.

Телемедицина - це спосіб надання медичної допомоги через інтернет, зазвичай за допомогою відеочату. Ця технологія має кілька переваг як для пацієнтів, так і для медичних працівників. Незважаючи на технічні труднощі та критику, телемедицина може доповнювати та покращувати загальний досвід пацієнтів.

Концепція телемедицини та пов'язаних з нею послуг тепер добре встановлена та вже підтверджена як корисна для суспільства. Вона забезпечує управління хронічними захворюваннями, дотримання рецептів, дистанційні послуги, допомогу в критичних та важких випадках тощо, що в кінцевому підсумку дозволяє цій методології підтримувати галузь охорони здоров'я та медичної допомоги. Крім того, низка телеречей для носіння допомагає пацієнтам та надає їм актуальну інформацію щодо стану їхнього здоров'я у незвичному способі.

Телемедицина є інноваційною технологією, і багато вважають її руйнівною інновацією. Тому, щоб обслуговувати віддалених пацієнтів, телемедицина використовує різноманітні засоби електронного зв'язку, від відеоконференцій до обміну зображеннями та дистанційного моніторингу

пацієнтів. Лікарі також можуть використовувати автоматизацію для надання якісного лікування своїм пацієнтам. Їм потрібно розробити кращі інформаційні системи підтримки та вивчити новий спосіб управління файлами. Наприклад, віртуальна консультація спонукає лікарів загальної практики звертатися до експертів, коли у них виникають питання щодо захворювання або лікування. Звіти про обстеження, історія, медичні висновки, рентгенограми або інші зображення надсилаються експерту для обстеження. Фахівець може відповісти електронно та організувати віртуальну зустріч з лікарем. Ці віртуальні консультації можуть усунути потребу в зайвих особистих направленнях до спеціалістів, скоротити час очікування на відгук від спеціалістів та усунути потребу в надмірних подорожах. Стратегії телемедицини особливо корисні там, де лікар може бачити пацієнта, встановлювати діагноз та фіксувати досвід.

Телемедицина є інноваційною технологією, і багато вважають її руйнівною інновацією. Таким чином, для надання послуг віддаленим пацієнтам, телемедицина використовує різні засоби електронного зв'язку, від відеоконференцій до обміну зображеннями та дистанційного моніторингу пацієнтів. Лікарі також можуть використовувати автоматизацію для надання якісного лікування своїм пацієнтам. Їм потрібно розробити кращі інформаційні системи підтримки та вивчити новий спосіб управління файлами. Наприклад, віртуальна зустріч спонукає лікарів загальної практики звертатися до експертів, коли вони мають питання щодо захворювання або лікування. Звіти про обстеження, історія, медичні результати, рентгенограми або інші зображення надсилаються експерту для обстеження, і фахівець може відповісти електронно та організувати віртуальну зустріч з лікарем. Ці віртуальні консультації можуть усунути необхідність в надмірних особистих направленнях до спеціалістів, скоротити час очікування відгуку від спеціалістів та усунути потребу в надмірних подорожах. Телемедицинські стратегії особливо корисні там, де лікар може бачити пацієнта, встановлювати діагноз та робити записи про лікування.

Електронна система особистих медичних записів використовує медичні записи, які можна відстежувати та зберігати. Ми можемо використовувати пристрій з доступом до Інтернету, такий як настільний комп'ютер або смартфон, у будь-який час. Особистий медичний запис швидко надасть необхідну інформацію рятувальникам в надзвичайних ситуаціях, таку як діагнози, рецепти, вподобання щодо препаратів та контактну інформацію лікаря. Застосування цієї технології розроблені для того, щоб допомогти клієнтам правильно організувати свої медичні записи в одному безпечному місці. Програми відновлення повинні встановлювати цілі для ефективного покращення поведінки пацієнтів, що стає простішим завдяки мобільним технологіям для охорони здоров'я. Вони дозволяють пацієнтам вести облік калорій, записувати важливі показники, фіксувати фізичну активність та контролювати дози та графіки прийому препаратів. Окрім діабету, ідея самостійного керування пацієнтом за допомогою телемедицини може бути розширена на інші тривалі хронічні захворювання, такі як гіпертонія та цілий ряд захворювань органів травлення.

Рішення в галузі телемедицини значно сприяють роботі лікарів. Проте, коли їх поєднати з штучним інтелектом (AI), вони можуть стати ще більш ефективними. Вони можуть спростити базові завдання, зменшити навантаження на лікарів і підвищити задоволеність роботою. Додаток передає дані вчасно та високої якості, щоб забезпечити гладкий хід консультації. Це дозволяє лікарям здійснювати більш ретельний аналіз стану здоров'я пацієнта та виявляти будь-які аномалії. Пацієнтів заохочується переглядати вільні години прийому лікаря, а також їхні майбутні зустрічі та можливість перепланувати їх. Аналітика в галузі охорони здоров'я часто використовується для взаємодії зі зібраними даними та їх перегляду через інтерфейс користувача. Зберігання та передача даних підвищують ефективність витрат часу та коштів в довгостроковій перспективі. Телемоніторинг базується на різних електронних технологіях, які передають

статистику пацієнта безпосередньо до аналітичного інтерфейсу медичного постачальника.

Ця технологія виявилася цінною і невід'ємною в різних сферах телемедицини та медичної промисловості. Вона помітно революціонує хірургічні операції, медичну підготовку та навчання. Після того, як лікар схвалив бажану зустріч, пацієнт повинен отримати підтвердження. Внутрішні записи, такі як огляди, зміни та важливі повідомлення з лікарні, можуть бути включені в їхні профілі. Лікарі можуть здійснювати зустрічі за допомогою розкладів. Розкладування та перепланування є стандартними функціями сучасних додатків для телемедицини. Коли лікарі інформовані про зустріч, вони можуть отримати доступ до медичного запису пацієнта та інших даних, які необхідні для належної консультації та діагностики.

Системи віртуальної реальності (VR) трансформували застосування в телемедичних пристроях, роблячи їх більш іммерсивними. У віртуальній реальності лікарі та їх команди тепер можуть спостерігати за операційними процедурами на 3D моніторах. За допомогою відеоконференцій, лікарі, і навіть хірурги, можуть працювати з пацієнтами, які знаходяться за тисячі миль від них. Це дозволяє навіть медичним командам із різних континентів співпрацювати та проводити відеоконференції у важливих та складних ситуаціях.

Платформи телемедицини можуть використовувати VR для симуляції даних пацієнта та створення графічного середовища з відеоконференціями для оптимізації комунікації з лікарем. Цей метод також корисний в сільських або віддалених регіонах і має вирішальне значення для надання медичних послуг на місцях. В сільських районах це сприяє збільшенню довіри до системи місцевої охорони здоров'я. Це також означає більше фінансування для місцевих служб охорони здоров'я для віддалених пацієнтів. Крім того, ця технологія дозволяє пацієнтам залишатися біля родини та друзів.

Телемедицина, порівняно зі звичайними методами лікування, має декілька потенційних недоліків. Вона не є заміною традиційної системи

охорони здоров'я; вона доповнює систему охорони здоров'я для обмежених функцій. Існує серйозна проблема взлому медичних даних пацієнтів, особливо якщо пацієнт підключається до телемедицини з громадської мережі або незашифрованого каналу. Коли людина потребує невідкладної допомоги, ця технологія може призвести до затримки в наданні медикаментозного лікування, особливо тому, що лікар не може надавати дистанційно невідкладну допомогу або проводити лабораторні тести. Правила держав відрізняються, і лікарі можуть бути не в змозі здійснювати медичну практику поза межами держави, в якій вони мають ліцензію, і держави, в якій проживає пацієнт. Клініки також повинні переконатися, що сервіс телемедицини, яким вони користуються, є безпечним і дотримується вимог щодо конфіденційності.

Під час сесій телемедицини провайдерам доводиться покладатися на самоповідомлення пацієнтів і вимагати від лікарів задавати додаткові запитання, щоб отримати повну історію хвороби пацієнта. Якщо пацієнт не повідомляє про важливі симптоми, які мали б бути виявлені під час особистого огляду, це може поставити під загрозу лікування. Одним із найважливіших недоліків є обмежена доступність і вартість. Це може бути витратним для постачальника налаштовувати та утримувати. Незважаючи на це, телемедицина може бути занадто дорогою для менших закладів охорони здоров'я. Погана комунікація також може ускладнити надання надійної медичної допомоги.

У майбутньому пацієнти будуть реєструватися на прийом до лікаря свого вибору після швидкої та простої реєстрації. Замість введення тексту, пацієнти будуть завантажувати свою медичну історію, документи для підтвердження, медичні звіти та минулі рецепти. Інтерфейс пацієнта це правильна функція, яка дозволяє лікарю створювати план невідкладної допомоги. Він відображає узагальнену інформацію про фізичний та особистий стан пацієнта, допомагаючи лікарю в прийнятті рішення. Місцеві ресурси охорони здоров'я можуть бути швидко мобілізовані для надання як невідкладних, так і не невідкладних послуг. Це дозволить клінікам

сконцентруватися на високопопитних складних випадках, замість невеликих випадків, які можна було б вирішити дистанційно.

У майбутньому відділення невідкладної медичної допомоги будуть використовувати відеоконсультації для триажування пацієнтів, що потребують невідкладної допомоги. У результаті цього менше відхилень від невідкладної допомоги та покращені стани пацієнтів. Кілька організацій охорони здоров'я впроваджують програмне забезпечення для телемедицини, щоб підготуватися до майбутнього. Воно забезпечує підключення кардіологів до пацієнтів, які приходять до сільського медичного закладу. Воно також скорочує час, необхідний для проведення контрольних візитів та доповнює звичайну систему. Дистанційне керування пацієнтами виступає як наступна велика тенденція в сфері віртуальної охорони здоров'я, і ця проблема створює значні можливості для багатьох компаній.

Телемедицина є цінною технологією, яка забезпечує зв'язок між лікарями та пацієнтами для забезпечення довгострокових змін у способі життя. Вона має значні переваги для персоналу медичного офісу. Це часто допомагає уникнути навантаження на реєстрацію пацієнтів і концентрується на вищій вартості завданнях. З можливістю онлайн-консультацій лікарі можуть доглядати за своїми пацієнтами, одночасно можливо надаючи допомогу іншим практикам. Це також зменшує відстані для обміну інформацією про діагноз, лікування та запобігання захворюванням між лікарем і пацієнтом за допомогою електронних засобів. Найширший спектр застосувань телемедицини може наблизити медичне обслуговування до людей, які проживають в сільських районах, де інакше доступ до якісної медичної допомоги був би обмежений. Останніми роками було показано, що ця технологія підвищує якість медичних послуг, дозволяючи обмін інформацією між віддаленими регіонами. Вона розширює доступ до служб для недообслуговуваних районів, полегшуючи їм записатися на прийом і утримати його. Люди з обмеженою мобільністю отримують консультації лікарів та рецепти, які вони потребують швидше. Ліки, тести та процедури, які їм слід

проводити, можна отримувати на своєму місці проживання. Телемедицина мінімізує подорожі лікарів і пацієнтів по всьому світу та змінює життя кожної хворої людини, забезпечуючи їй належне медичне обслуговування.

1.2 Складові частин телемедицини

Як і традиційна медицина, телемедицина складається з багатьох частин, кожна з яких має свої особливості. І як в медицині важко поняти грань коли один вид лікування переходить в інший. Думки з цього приводу розділяються в багатьох експертів, одні розділяють телемедицину на сім і більше незалежних процедур, а інші схилиються до трьох, або чотирьох. Я виділив шість аспектів які на мою думку найважливіші.

1.2.1 Телемедичне консультування

Найпоширенішою процедурою сьогодні є телемедична консультація (дистанційне обговорення клінічного випадку), що дає наближення кваліфікованої допомоги, своєчасна підтримка для прийняття клінічних рішень та покращення якості та доступність медичної допомоги. Надати телемедичну консультацію є невід'ємна частина щоденної лікувально-діагностичної роботи. Для отримання підтримки прийняття найкращих клінічних рішень, які повинен приймати кожен лікар вміння готувати медичну інформацію та проведення телемедичного консультування. Менеджер системи охорони здоров'я повинні також володіти цими навичками, щоб підтримувати свою лікувальну роботу, аудит телемедичної діяльності, інформаційний контроль безпечно забезпечити надання медичних послуг на сучасному рівні. Це обов'язковий компонент діяльності сучасного лікаря полягає в спілкуванні з колегами через Інтернет (тематичні списки розсилання, професійні Інтернет-товариства, соціальні мережі), задля ефективнішого вирішення поставленої

проблеми, що також у тому числі неформальні дистанційні консультації, які можуть проходити без узгодження, Але відповідає всім вимогам якості і безпеки.

Телемедичні консультації в основному базуються на різноманітних відеоконференції, відправка електронних листів і навіть телефоні звінки. Тому ця тенденція настільки популярна в наш час, тай в Україні.

Під час вибору телемедичних консультацій звертається особливий акцент на конфіденційності інформації, професійності лікарів та безпеку.

Отже, слід зусереджувати увагу на профусійності лікаря і не менш важливо на захисті даних, що знаходяться безпосередньо в системі, залишивши на другий план швидкість та безперервність каналу. Треба розуміт, що відповіді моментальної на ваші листи можить і не бути, а відео звінки не є пріорітетним вид діяльності роботи лікарів на робочому місті.

1.2.2 Біотелеметрія

Телеметрія - сукупність технічних засобів і методів вимірювання відстані між різними фізичними об'єктами, технологічними та іншими значеннями в промисловості, енергетичних, транспорт та інші об'єкти. Передача певних даних з будь-якого місця на віддалений термінал.

Телеметрія в контексті телемедицини (фізіологічна телефіксація параметрів) з'явилися як невід'ємна частина космічної медицини, а її цивільні різновиди телемоніторинг для інтенсивної терапії і транспортування важкохворих пацієнтів. Використовується для військової телеметрії, аерокосмічні медицині та медицині катастроф. Найпоширеніша форма клінічного застосування телеметрії є теле-ЕКГ.

По-перше, процедура спрямована на діагностику стану пацієнта, де основним факторами є точність і швидкість роботи обладнання. Основним фактором таких процедур є точність, достовірність, безперервність і швидкість.

По-друге конфіденційність і можливість віддаленого моніторингу системи.

Як згадувалося раніше, теле-ЕКГ є найпоширенішою формою і став першим прикладом телемедицини в світі.

Система підтримує живучість за допомогою набору програм, що поєднує в собі датчики та термінали, які виводять дані. Основна складова такої системи є саме апаратне забезпечення, оскільки програмний комплекс простий, і грають менш важливу роль.

Медицину катастроф також слід вважати незалежною частиною біотелеметрії, військова біотелеметрія та аерокосмічна медицина, де відстані вже зайняті одним із перших пунктів списку пріоритетів, поряд з надійністю та точністю. Дана система оснащена складним комплексом програм, однаково складних являє собою технічна частина, що являє собою результат багатьох професіоналів і великих робіт це не тільки потребує часу, але й коштує грошей.

1.2.3 Домашня(персональна) телемедицина

Завдяки домашній телемедицині вона дозволяє медичним працівникам контролювати стан пацієнта за допомогою різноманітних медичних пристроїв, які можна встановити вдома, щоб контролювати життєві показники пацієнта та частіше отримувати оновлення про стан здоров'я пацієнта. Дистанційний моніторинг особливо корисний для пацієнтів із хронічними захворюваннями або пацієнтів, які одужують після серйозних травм, хвороби чи операції, лежачих пацієнтів та пацієнтів похилого віку, стан здоров'я яких не дозволяє їм звернутися до лікаря.

Навіть якщо порівнювати з іншими процедурами телемедицини, домашня або персональна телемедицина – це досить молода галузь, яка стрімко розвивається і набуває популярності в наш час. Телесестринство,

телеконференції, телеконсультації, навіть деякі аспекти біотелеметрії – все це входить в рамки домашньої телемедицини.

Прежде всього, ця процедура орієнтована на людей, які з різних причин не можуть перебувати в лікарні. Це можуть бути політики, знаменитості різних сфер, серйозно хворі пацієнти, або навіть літні люди, які прагнуть провести залишок свого життя вдома, поруч з родиною, а не в лікарні. Не слід також забувати, що хоча персональна телемедицина спочатку створювалася для цільових груп, вона з часом стала все більш популярною і доступною завдяки розвитку технологій.

Варто відзначити, що в наш час навіть представники середнього класу можуть дозволити собі придбати комп'ютер і скористатися особистою телемедициною.

Основною перевагою цієї процедури є конфіденційність, точність і доступність. Оскільки програма повинна бути доступною цілодобово, вона повинна забезпечувати необхідний рівень медичної допомоги та відповідати високим медичним стандартам.

1.2.4 Телескринінг

Телескринінг, що означає дистанційне виявлення та формування груп ризику, а також впровадження запобіжних заходів з використанням телемедицини, надає широкий спектр комплексних профілактичних заходів. Особливий акцент робиться на покращенні здоров'я дітей та підлітків, зокрема для немовлят із вадами зору, а також тих, хто має порушення постави у підлітків та інші проблеми.

Телескринінг виявляється особливо ефективним при ранньому виявленні онкологічних та фтизіатричних захворювань. Такий підхід спрямований на покращення здоров'я та виявлення ризикованих груп серед сільського населення та закритих груп людей.

Метою цієї програми є покращення профілактичних заходів рагального медичного рівня.

Скринінг спрямований на раннє виявлення і діагностику захворювання, на ранній стадії коли пацієнти перебуває далеко від експерта, який може провести належне обстеження і поставити діагноз.

Дана процедура забезпечується за допомогою цілого комплексу, перш за все, апаратного. Окрім забезпечення необхідного обладнання для сканування, для телескринінг також потрібен пакет, який також поділяється на компоненти. Першим таким компонентом є база даних, в якій зберігається вся необхідна інформація. Другим – програми, що проводять первинну вибірку, і схвалено відповідними експертами.

Тут конфіденційність пацієнта є необхідною складовою, але першою за все має бути точність та актуальність. Формування груп ризику та їх подальше спостереження дослідження відповідних експертів – час і гроші, а іноді й життя. Програма має працювати точно, інформація, що циркулює в ньому, повинна бути повною, а база даних зберігати свою структуру та дані, навіть якщо інші системи виходять з ладу.

1.2.5 Телеприсутність

У 1995 році вперше було запропоноване поняття «телеприсутності» у сфері військової медицини. Телеприсутність – це віртуальна присутність травматолога або хірурга, що надає якісну та спеціалізовану допомогу пацієнтам з травмами та ортопедичними захворюваннями. Ця телемедична процедура досягається використанням комп'ютерних систем, бездротових телемедичних систем, комп'ютерних мереж з базами даних, систем медичної розвідки, обміну змінною інформацією в реальному часі, систем віртуальної реальності. У 2002-2004 р. була розроблена усталена повноцінна концепція телеприсутності в травматології та ортопедії. Телеприсутність означає безперервну дистанційну участь фахівців у лікуванні та діагностиці пацієнтів

у важких і критичних ситуаціях. При цьому пацієнти можуть проходити лікування в будь-якому медичному закладі (особливо такі системи асоціюються з сільськими лікарнями, важкодоступними та віддаленими районами).

1.2.6 Дистанційне навчання

Дистанційне навчання, яке використовує комп'ютерні та телекомунікаційні технології, є формою освіти, спрямованою на інтерактивну взаємодію між викладачем та студентами на різних етапах навчання, а також на самостійну роботу з матеріалами інформаційної мережі.

В сучасному світі Інтернет-технології глибоко впливають на всі сфери нашого життя, включаючи освіту, що прискорює розвиток інформаційного суспільства та подолає географічні обмеження. Освіта стає більш доступною, оскільки вам вже не потрібно фізично знаходитися біля вчителя. Хоча форма заочного навчання існує тривалий час, її можливості обмежені. Інтернет дозволяє розширити ці можливості, роблячи дистанційне навчання повніше та всеосяжним.

Нині, в умовах економічних відносин та жорсткої конкуренції на ринку праці, знання, навички та досвід набувають особливого значення. Фахівець 21 століття – це людина, яка вільно користується сучасними інформаційними технологіями, постійно підвищує та вдосконалює свій професійний рівень. Отримання нових знань і навичок, які є практично корисними та використовуються в роботі інформаційного суспільства, значно розширює можливості самореалізації та сприяє кар'єрному розвитку. Однак однією з головних перешкод для тих, хто бажає продовжити навчання (з огляду на те, що більшість із них вже працює), є брак часу. Більшість не можуть відвідувати школу щодня. Відстань – ще одна серйозна перешкода. Якщо школа знаходиться в іншому місті, відвідувати заняття часто незручно та дорого, класична заочна форма навчання часто не виправдовує своєї мети. Знання, які

отримує учень, часто є поверхневими, а самі уроки непродуктивними. До того ж процес навчання займає досить тривалий час.

1.3 Технічний аналіз багатосервісної волоконно-оптичної мережі для телемедицини.

Широкопasmові мережі стають необхіднішими завдяки високошвидкісному доступу до Інтернету, телемедицині, відеоконференціям, дистанційному навчанню та іншим передовим сервісам. Сучасні комп'ютери використовують ці мережі для отримання доступу до різноманітних послуг і взаємодії з розмаїттям сервісів, які надаються в цьому інфраструктурному середовищі. З метою ефективного забезпечення високої пропускної здатності в широкопasmових мережах широкого використання використовується концепція оптичного доступу FTTx (Fiber To The x). Ця концепція охоплює різноманітні варіації, такі як FTTB (Fiber To The Building) і FTTH (Fiber To The Home), а також FITB (Fiber To The Building) і технологію гібридного доступу HFC (Hybrid Fiber Coaxial). Подальший розвиток сучасних телемедицинських мереж, які передусім представлені корпоративними мережами оптичного доступу до Інтернету, також сильно залежить від вдосконалення регіональних волоконно-оптичних ліній зв'язку.

Мережі мультисервісного характеру для телемедицини, подібно до інших розподільних оптичних мереж доступу, можуть базуватися на технології PON (Passive Optical Networking) або OE (Optical Ethernet). Перша з цих технологій ґрунтується на використанні стандартів швидкісних волоконно-оптичних каналів (Fibre Channel) і FDDI (Fiber Distributed Data Interface), роблячи телемедицинські послуги ще більш доступними та ефективними.

Fibre Channel - це визнаний стандарт протоколів для високошвидкісної передачі даних, і його основне використання знаходиться в мережах зберігання даних. Цей клас протоколів належить до технічного комітету T11

та має міжнародне визнання у сфері інформаційних технологій. Почавши як технологія для суперкомп'ютерів, Fibre Channel успішно перейшов до сегмента мереж зберігання даних, де встановився як стандартний інтерфейс для з'єднання з системами зберігання. Його перевагами є висока швидкість передачі даних та значна дальність, що робить його ефективним в розрізі забезпечення зв'язку в системах зберігання даних. Важливо відзначити, що використання одного волоконно-оптичного каналу для створення топології мереж на основі Fibre Channel дозволяє економно використовувати ресурси і сприяє оптимізації інфраструктури (рис. 1.2).

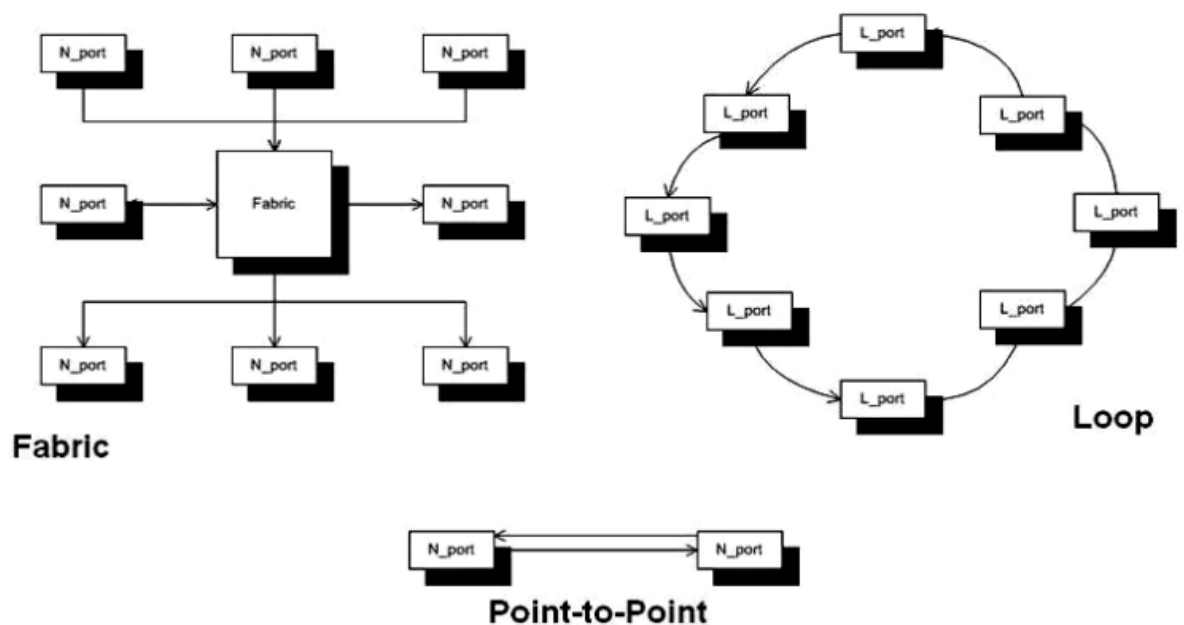


Рисунок 1.2 – Топології стандарту Fibre Channel (FC): fabric (комутована зв'язна), loop (керована петля) та point-to-point (точка-точка).

Недоліком Fibre Channel (FC) є відсутність енергетичної автономності та регенератора на області волоконно-оптичних ліній зв'язку, що призводить до зниження стабільності його застосування на високошвидкісних магістральних каналах мереж оптичного доступу.

Fiber Distributed Data Interface (FDDI) – це волоконно-оптичний канал передачі даних у розподілених інтегрованих мережах (див. рис. 1.2), який забезпечує передачу інформації зі швидкістю від 100 Мбіт/с до 1 Гбіт/с для

багатомодового оптоволоконна та 10-80 Гбіт/с для одномодового оптоволоконна на довжинах хвиль 1310 нм та 1550 нм. Дальність передачі інформації у межах 40 км, а для експериментальних лабораторних зразків – до 120 км. FDDI визначається високою швидкістю передачі даних та значною дальністю, роблячи його оптимальним вибором для розподілених інфраструктурних систем.

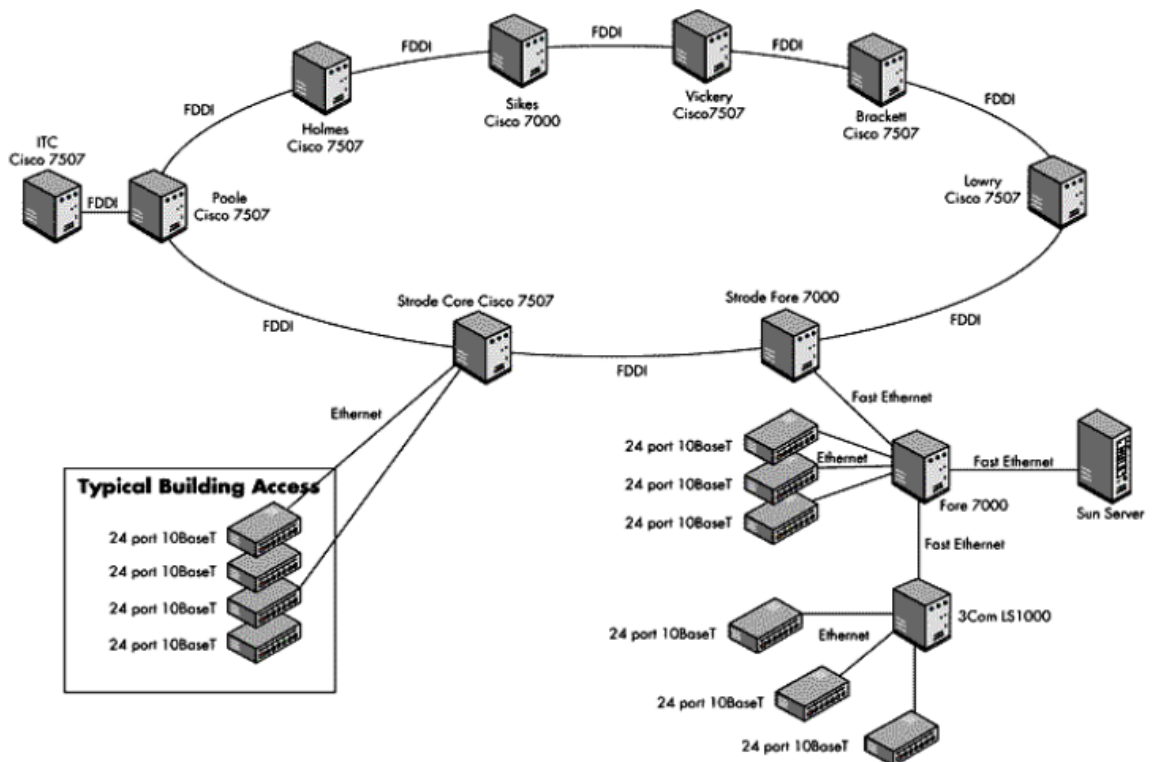


Рисунок 1.2 – Варіант кільцевої топології мультисервісної мережі доступу на базі стандарту FDDI.

Волоконно-оптичний стандарт каналів даних FDDI рівень програмного забезпечення являє собою набір протоколів ANSI для цифрової передачі даних по волоконно-оптичній лінії. Інформаційна мережа FDDI (подібна до стандарт IEEE 802.5) відповідно до протоколу Token Ring (подвійне кільце) Підтримує швидкість передачі даних до 100 Мбіт/с - 12 Гбіт/с в мережі. Інформаційні мережі на основі FDDI часто використовуються як автомагістралі через підтримку високої пропускної здатності і великих ретрансляційних ділянок.

Перевагами FDDI є високий захист від перешкод, конфіденційність передачі інформації та відмінна гальванічна розв'язка користувача. Високі швидкості передачі, які легше досягти за допомогою волоконно-оптичних

кабелів, дозволяють вирішити багато проблем, які не можуть мати низькошвидкісні мережі, наприклад, передача зображень у режимі реального часу. Крім того, волоконно-оптичні кабелі легко вирішують проблему передачі даних без ретрансляторів на кілька кілометрів, дозволяючи будувати великі мережі, навіть охоплюючи цілі міста, з усіма перевагами локальних мереж (включаючи низький рівень бітових помилок). І хоча обладнання FDDI поки не популярне, перспектива дуже хороша. Основним недоліком, пов'язаним із застосуванням у високостабільних ІМ є недостатня енергетична автономність вузлів і проміжного обладнання.

PON (Passive Optical Network) – це мережа, яка базується на принципах пасивної оптики. Ця технологія дозволяє створити розгалужену мережу, в основному маючи деревоподібну топологію, із застосуванням виключно пасивних оптичних розгалужувачів.

Технологія PON виявляє особливу ефективність при охопленні обширних територій з різною густотою забудови - від багатоквартирних будинків до сільських поселень і міст, де переваги цієї технології наочно проявляються. Передача і отримання сигналів в обидва напрямки, як правило, здійснюються через одне оптичне волокно, проте при різних довжинах хвиль (1310 і 1490 нм).

Інформація про всіх користувачів передається одночасно з використанням розподілу часу каналу, починаючи від головної станції - оптичного терміналу (OLT, термінал оптичної лінії) - та закінчуючи кінцевим модулем оптичної мережі (ONU, оптичний мережевий блок). Оптична потужність на виході OLT розподіляється (рівномірно або нерівномірно) таким чином, щоб рівень сигналу на вході всіх ONU був приблизно однаковим. Optical Ethernet - набір стандартів Ethernet для волоконно-оптичних каналів передачі даних (FE/GE/10GE/40GE/100GE) для побудова локальної мережі LAN та локальної обчислювальної мережі MAN і забезпечує швидкість передачі даних від 100 Мбіт/с до 10 Гбіт/с на відстані 1 До 10 км при 1550 для систем з одномодовим оптичним волокном. Надає остання версія інтерфейсу

100GE швидкість до 100 Гбіт/с і відстань до 70 км без ретрансляторів і підсилення і є одним з основних учасників глобальної мережі при побудові магістралей глобальних мереж.

Усі відомі топології волоконної мережі використовуються для Побудова ієрархії комунікаційної мережі, підключеної до глобальної мережі Інтернет, який можна використовувати для організації телемедичних каналів інформаційна мережа.

Порівняльний аналіз параметрів зазначених стандартів для мультисервісних телемедичних волоконно-оптичних мереж доступу представлено в таблиці 1.1.

Таблиця 1.1 включає параметри сучасних каналів ВОЛЗ для комп'ютерних мереж

Волоконно-оптичний канал	Кількість каналів	Максимальна пропускна здатність, Мбіт/с	Максимальна дальність передачі даних, км
Fibre Channel	1	100-28050	50
FDDI	1	100-12000	70
Optical Ethernet (FE/GE/10GE/40GE/100GE)	2	100-100 000	40

Серед провідних світових виробників апаратних і програмних рішень для телемедичних інформаційних мереж волоконного зв'язку виділяються такі компанії, як Cisco Systems Inc., Nortel Networks, OptiStar, Alkatel, 3Com Corp., Bay Networks Inc., VLSI Technology, Siemens, Lucent, Marconi Communications та інші визначені гравці ринку.

1.4 Архівація даних

Архівація даних є ключовим етапом у обробці інформації, використовуючи різноманітні техніки для упакування та стиснення інформації. Головна мета цього процесу - оптимізація обсягу зберігання та

зменшення часу передачі даних, що стає критичним для сучасних систем обробки інформації.

Перший важливий аспект архівації - це стиснення даних, що використовує різні алгоритми для зменшення обсягу інформації. Цей процес може бути безвартним або втратним, залежно від конкретних вимог системи. У архіваторах також можуть бути вбудовані механізми корекції помилок для забезпечення надійності даних.

Другий аспект - захист від несанкціонованого доступу, реалізований за допомогою парольного захисту. Це важливо для забезпечення конфіденційності даних, особливо при їх передачі через відкриті мережі.

Дослідження також охоплює вибір оптимальних форматів архівів. Різноманітність форматів, таких як ZIP, RAR, 7z, розширює можливості використання технології архівації в різних сценаріях.

У контексті дипломної роботи особливий акцент приділяється важливості архівації для оптимізації передачі та зберігання даних. Вивчаються вплив архівації на швидкість передачі та можливість використання різних алгоритмів стиснення враховуючи вимоги конкретного додатку чи середовища.

Додатково, важливим елементом архівації є здатність системи виявляти та виправляти можливі помилки під час розпакування даних. Це має велике значення в забезпеченні інтегритету інформації під час передачі та зберігання, особливо в умовах нестабільних мереж чи при зберіганні на носіях, які можуть піддаватися впливу зовнішніх факторів.

Важливо відзначити, що архівація даних є не лише засобом економії місця чи прискорення передачі, але також сприяє забезпеченню конфіденційності, цілісності та доступності інформації. Це робить архівацію ключовим елементом сучасних інформаційних технологій, особливо в умовах зростаючого обсягу даних та потреби в їх швидкому та безпечному обробленні.

Алгоритм Хаффмана, визнаний ефективним методом стиснення даних, базується на ідеї надання коротших бітових кодів частішим символам. В контексті архівації, він використовується для безвтратного стиснення, де частотність входження символів визначає їх бітове представлення. Такий підхід має суттєві переваги в оптимізації обсягу даних для подальшої передачі чи зберігання.

Поєднання алгоритму Хаффмана та архівації даних дозволяє досягти ефективного використання простору зберігання. Символи, які зустрічаються частіше, отримують коротші кодові слова, що сприяє зменшенню обсягу інформації. Адаптивність алгоритму робить його відмінним інструментом для оптимізації різноманітних типів даних.

Поєднання алгоритму Хаффмана та архівації даних є важливим кроком в напрямку оптимізації передачі та зберігання інформації. Зменшення обсягу даних допомагає прискорити їх передачу через мережу, особливо в умовах обмеженої пропускної здатності. Такий підхід стає ключовим для оптимізації обробки та зберігання інформації в умовах зростаючого обсягу генерованих даних.

Враховуючи сучасні вимоги до обробки та обміну даними, використання алгоритму Хаффмана в архівації стає важливим елементом технологічного ландшафту. Його ефективність та адаптивність роблять його необхідним інструментом для оптимізації даних у сучасному інформаційному середовищі.

2 ОПИС ОСНОВНИХ ПОНЯТЬ У СТИСНЕННІ ДАНИХ З МЕТОЮ ОПТИМІЗАЦІЇ ПРОЦЕСУ ПЕРЕДАЧІ ДАНИХ

2.1. Ентропія

Для оцінки ефективності алгоритму стиснення одним із ключових методів є розрахунок середньої кількості бітів, необхідних для кодування кожного символу. Теорія інформації надає математичний каркас, який допомагає зрозуміти передачу повідомлень через системи зв'язку. Вона охоплює всі процеси, від усунення зовнішнього шуму до оптимального кодування подій, що відбуваються під час переходу від вхідних повідомлень до вихідних кодів та виявлення помилок.

Теорія інформації дає можливість кількісно вимірювати обсяг інформації, який міститься в отриманих даних. У своїй статті "Математична теорія зв'язку" Клод Шеннон досліджував основні концепції цієї теорії. Вивчаючи, скільки інформації можна передати через канали зв'язку, Шеннон визначив, що чим менше ймовірність події, тим більш інформативною вона стає. Ентропія є одним із ключових показників цієї теорії і визначає ступінь невизначеності. Шеннон позичив цю концепцію з термодинаміки, замінивши "стан" на "повідомлення".

Ентропія відіграє важливу роль у оцінці ефективності передачі інформації через канали зв'язку, оскільки вона вказує на мінімальну кількість бітів, які в середньому потрібні для кодування конкретного потоку даних.

Стиснення - це процес представлення даних за допомогою найменшої кількості бітів. Ця технологія зменшує необхідний обсяг пам'яті і збільшує можливість передачі даних. Метою зменшення надлишкової інформації в збережених і переданих даних є підвищення ефективності, оптимізація використання ресурсів і покращення управління даними. Надлишковість - це дублікати або повторювані дані в наборі даних або передачі.

Методи стиснення дозволяють зберігати дані більш компактно, зменшуючи необхідний простір для зберігання. Стиснені дані потребують меншої кількості бітів для передачі під час передачі, що скорочує час передачі і зменшує використання пропускної здатності. Стиснення має вирішальне значення для оптимізації ємності сховища, підвищення ефективності передачі даних і прискорення обробки великих наборів даних [1].

Завдяки стисненню даних зменшується обсяг необхідного простору для зберігання. Це особливо вигідно для великих наборів даних, файлів або архівів. Методи стиснення усувають надмірність, більш ефективно кодують шаблони і мінімізують загальний розмір даних. Це призводить до значної економії місця на пристроях зберігання або серверах.

Стиснення може значно скоротити час передачі даних при передачі даних через мережі або Інтернет. Стиснені дані потребують менше бітів для представлення тієї самої інформації, що призводить до меншого розміру файлів. Файли меншого розміру можна передавати швидше, особливо в умовах обмеженої пропускної здатності або при передачі великих обсягів даних. Швидша передача може підвищити ефективність, зменшити затримки та покращити користувацький досвід.

Методи стиснення дозволяють зберігати дані більш компактно, зменшуючи необхідний простір для зберігання. Стиснені дані потребують меншої кількості бітів для передачі під час передачі, що скорочує час передачі і зменшує використання пропускної здатності. Стиснення має вирішальне значення для оптимізації ємності сховища, підвищення ефективності передачі даних і прискорення обробки великих наборів даних.

Стиснення поділяється на дві категорії залежно від необхідного рівня відновлення: стиснення без втрат і стиснення з втратами. Стиснення без втрат призводить до ідентичної реконструкції вихідних даних, тоді як стиснення з втратами створює реконструйовані дані, які відрізняються від вихідних. Стиснення з втратами часто використовується в інтернет-додатках, особливо для потокового мультимедіа та телефонії. Зменшення кількості даних,

необхідних для представлення тексту, називається стисненням тексту. Ця техніка є важливою для стиснення без втрат, оскільки відновлений текст має бути таким самим, як і оригінальний вхідний. Будь-які незначні відмінності у відновленому тексті можуть призвести до зміни його змісту. Стиснення тексту можна досягти за допомогою різних алгоритмів статистичного моделювання, включаючи перетворення Берроуза-Вілера [2], передбачення за частковим збігом, Лемпеля-Зіва-1977 та Лемпеля-Зіва-Велча. Ступінь стиснення можна визначити за допомогою різних алгоритмів, деякі з яких зберігають вихідну інформацію і вважаються алгоритмами без втрат.

На відміну від них, інші можуть спричинити втрату певної інформації під час стиснення. Важливо знати, що кожен метод стиснення спеціально розроблений для певного типу зображень і може не давати оптимальних результатів для інших зображень. Деякі алгоритми можуть змінювати змінні для покращення результатів стиснення і отримання більш якісного зображення. Шифрування захищає інформацію та зберігає її конфіденційність, роблячи дані нечитабельними для сторонніх осіб. Воно має вирішальне значення для захисту конфіденційної інформації, безпеки комунікації, дотримання нормативних вимог, побудови довіри та забезпечення цілісності даних. Шифрування має важливе значення для сучасних стратегій інформаційної безпеки і допомагає зменшити ризики, пов'язані з несанкціонованим доступом і витоком даних [3].

Заходи контролю секретних ключів є важливими для безпеки стиснутих і зашифрованих даних. Вони є важливими даними для збереження конфіденційності та цілісності даних.

Організації можуть підвищити безпеку стиснутих і зашифрованих даних, впровадивши надійні заходи контролю секретних ключів. Ці заходи захищають від несанкціонованого доступу до конфіденційної інформації, мінімізують ризик компрометації ключів і забезпечують конфіденційність і цілісність даних протягом усього їхнього життєвого циклу. Ефективні заходи

контролю ключів мають вирішальне значення для загальної стійкості та надійності систем шифрування.

Щоб захистити повідомлення, доцільно використовувати криптографію перед його приховуванням. Широко використовуються декілька криптографічних алгоритмів, включаючи алгоритм Рівеста-Шаміра-Адлемана, алгоритм Advanced Encryption Standard, Blowfish та Data Encryption Standard [4,5].

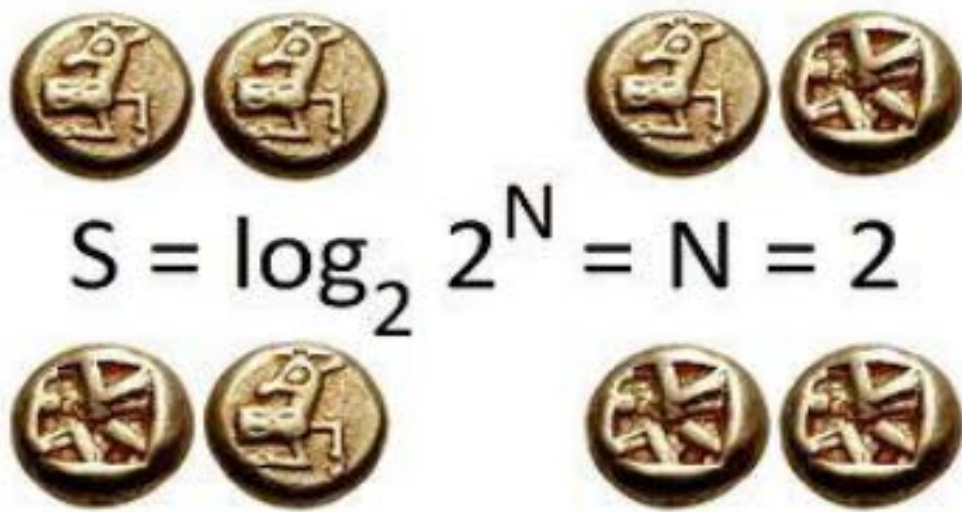


Рисунок 2.1 – Приклад ентропії для монетках

Коли кидаєш дві монети, можна застосувати ідею ентропії. Ентропія тут визначає логарифм за основою 2 кількості можливих результатів, еквівалентного двійковому біту. Таким чином, для опису чотирьох можливих результатів використовуються два біти ентропії.

Шифрування забезпечує приватність і конфіденційність даних. Шифрування захищає дані від зловмисника, який не може прочитати інформацію. Стиснення даних - це метод, який усуває повторювані рядки символів у файлі. Результатом цього процесу є стиснутий файл з більш рівномірно розподіленими символами [6]. Як наслідок, стиснений файл потребує менше часу для шифрування, розшифрування та передачі, що призводить до скорочення відкритого та зашифрованого тексту. Стандартний

алгоритм шифрування і стиснення виконуються послідовно, і для забезпечення конфіденційності їм надається перевага перед стиснутими даними [7]. Для ілюстрації, колекція файлів може бути об'єднана в архів WinZip, кожен з яких проходить свій власний процес стиснення і шифрування [8]. Символи зашифрованого тексту повинні мати майже рівномірний розподіл, що забезпечується хорошим алгоритмом шифрування. Алгоритми стиснення знаходять і видаляють повторювану або надлишкову інформацію в наборі даних, що призводить до більш мініатюрного представлення даних. Однак алгоритми шифрування покладаються на випадковість і непередбачуваність даних, щоб забезпечити їхню конфіденційність і цілісність. Шифрування стиснутих даних може призвести до втрати випадковості та введення шаблонів, які можуть полегшити зловмиснику розшифрування даних.

Під час стиснення даних алгоритм стиснення створює стиснене представлення, яке має менший розмір, ніж оригінальні дані. Таке зменшення розміру може розкрити інформацію про структуру або вміст даних, що може допомогти зловмиснику зламати шифрування. Щоб вирішити цю проблему, рекомендується застосовувати шифрування перед стисненням. Якщо спочатку зашифрувати дані, а потім стиснути зашифровані дані, алгоритм стиснення не зможе використати шаблони або надлишковість у даних, оскільки вони виглядатимуть як випадкові дані. Такий підхід гарантує, що процес стиснення не ставить під загрозу безпеку зашифрованих даних.

Одним з ефективних методів досягнення високого ступеня секретності є використання підходу "стиснути, а потім зашифрувати". Важливо зазначити, що згаданий підхід має суттєвий недолік - він міг би бути більш гнучким при аналізі стиснутих даних на предмет виявлення певних закономірностей [9]. Шифрування може ускладнити обробку стиснутих даних для пошуку тексту, що обмежує потенційні варіанти використання. Стиснення даних передусім має на меті зменшити обсяг пам'яті або час передачі, тоді як криптографія захищає дані. Шифрування і стиснення - це дві основні функції, коли дані

передаються через Інтернет або іншим незахищеним каналом. Шифрування забезпечує приватність і конфіденційність даних. Символи зашифрованого тексту повинні мати майже рівномірний розподіл, що забезпечується хорошим алгоритмом шифрування. У деяких випадках шифрування перед стисненням може бути кращим, особливо при передачі надлишкової інформації обмеженим і незахищеним каналом зв'язку. Такий підхід забезпечує конфіденційність даних, що передаються, а також зберігає пропускну здатність каналу зв'язку.

На схемі, зображеній на рис. 1, представлений універсальний метод, що поєднує стиснення і шифрування даних в одному кроці. Це вважається другим методом забезпечення безпеки стисненої інформації. Цей спільний підхід вирішує дві проблеми при передачі конфіденційних даних через мережу: швидкість і безпека. Поєднання цих двох процесів в одному кроці ефективно вирішує як завдання стиснення, так і завдання безпеки. Скремблювання вхідного тексту шляхом перестановки, перетворення Берроуза-Вілера (BWT) [2], переміщення вперед (Move-to-Front, MTF) і кодування довжиною пробігу можна комбінувати для досягнення ефективного стиснення даних. У поєднанні ці методи створюють надійну схему стиснення. Спочатку текст скремблюється, щоб внести випадковість і порушити шаблони. Потім послідовно застосовуються перестановка, BWT, MTF і кодування довжини потоку для подальшого зменшення розміру даних. Ця комбінація дозволяє ефективно стискати дані, зберігаючи при цьому їх цілісність і конфіденційність.

Підхід до безпечного та ефективного стиснення заснований на використанні скремблірованого входу. Це досягається шляхом генерації ключа і застосування перестановки. Як значення ключа використовується частота символів у вхідному тексті. Запропонований метод включає в себе скремблювання вхідних даних за допомогою згенерованого ключа, перетворення за допомогою BWT та стиснення за допомогою кодування Move-To-Front та Run-Length Encoding.

Теорема Шеннона про кодування джерела встановлює максимальний ліміт стиснення даних та числову ентропію. Відома також як теорема про безвтратне кодування, ця теорема визначає можливість ефективного стиснення даних без втрат, враховуючи їхню ентропію та інші фактори. Для випадкової величини N , де джерело X має ентропію $H(x)$, можна стиснути дані приблизно на $N * H(x)$ бітів без втрати інформації. Однак при наближенні значення N до нескінченності існує ймовірність втрати даних.

Ентропія даних залежить від ймовірностей виникнення подій і досягає свого максимального значення, коли ймовірності всіх конкретних подій рівні. Вона може мати лише додатне значення.

Допустимий ступінь стиснення даних:

$$\mu = \frac{H(x)}{H_{max}(x)} \quad (3)$$

де, H_{max} – максимальне значення ентропії.

У випадку максимальної ентропії для певного алфавіту необхідна більша кількість символів для передачі однакового обсягу інформації:

$$n_{min} = \mu n \quad (4)$$

де, n – кількість символів

Формула середньої ентропії повідомлення, або ентропія джерела у заданий момент часу:

$$H = - \sum_1^n P_i \log_2 P_i \quad (5)$$

де, H – середня кількість біт на одиницю інформації, P_i – ймовірність появи події, n – кількість подій.

Отже, коли ймовірність виникнення певної події складає P_i , оптимальним буде представити цю подію у вигляді $\log_2 P_i$ бітів. У випадку, коли кожен символ у повідомленні кодується за однаковою кількістю бітів, ми досягнемо мінімальної загальної довжини стиснутого повідомлення.

Цей принцип використовується у теорії інформації для оптимального кодування, де кожен символ асоціюється з відповідною ймовірністю та використовується необхідна кількість бітів для його кодування. Такий підхід забезпечує ефективне стиснення повідомлення, зменшуючи його загальний обсяг при збереженні важливої інформації.

Середня довжина кодів:

$$H = -\sum_1^n P_k H_k \quad (6)$$

де, P_k – ймовірність що джерело знаходиться у стані k .

Зі збільшенням кількості повідомлень ентропія також зростає, оскільки ймовірності змінюються, що викликає збільшення рівня невизначеності.

Для визначення надлишковості даних застосовується формула, що визначається як різниця між ентропією, яку можна було б досягти в ідеальних умовах, та фактичною ентропією, враховуючи конкретні обставини в даному випадку.

Ця формула допомагає визначити ступінь, до якої інформація у повідомленні може бути оптимально стиснутою або, навпаки, наскільки її варто деталізувати. У міру збільшення кількості повідомлень, надлишковість може виявитися важливим параметром для ефективного управління інформацією та її стиснення.

$$R = [-\sum_1^n P \log_2 P] - [-\sum_1^n P_i \log_2 P_i] \quad (7)$$

Отже, у випадку відсутності надмірності, вираз приймає такий вигляд:

$$\log_2 n - [-\sum_1^n P_i \log_2 P_i] = 0 \quad (8)$$

Засоби стиснення даних без втрат не здатні оптимально працювати для різноманітних видів інформації, відповідно до принципу Діріхле. Цей принцип визначає, що існує необхідність вибору підходів до компресії враховуючи конкретний тип даних та його особливості.

Якщо застосувати до компресії принцип Діріхле, де алгоритм стискає деякі дані, то відзначається те, що в процесі стиснення певні елементи можуть втратити свою деталізацію, в той час як інші можуть бути розширені для забезпечення ефективності стискання.

Це дозволяє адаптувати процес стиснення до конкретних характеристик даних, що сприяє оптимальному використанню алгоритмів компресії.

Також існує лема про накачку, яка формулюється в оборотному вигляді:

"При розміщенні n елементів у k ящиках завжди існує такий ящик, що містить максимум n/k елементів."

Давайте розглянемо конкретний приклад:

Уявімо послідовність довжиною n , де $n = 2^p$, а також послідовність 2^{pr} з довжиною pr , де pr - це довжина pr . Розглянемо випадок, коли довжина становить 3 біти (2^3), що утворює 8 послідовностей: 000, 001, 010, 011, 100, 101, 110, 111. Після застосування методу Ultrazip до цієї послідовності всі довжини зменшуються до лише 2 біт. Таким чином, для даної послідовності коефіцієнт стиснення, про який ми розповімо пізніше, буде більший за одиницю.

Якщо розглянути послідовність розміром 2^3 , отримаємо 4 послідовності: 00, 01, 10, 11. Зараз нам потрібно лише 2 біти, але після використання Ultrazip, принаймні одна з цих послідовностей займе три біти.

Оскільки перший приклад із довжиною 3 більший за другий приклад, програма r повинна взяти кілька послідовностей з набору A і включити їх у набір B . Це стає можливим завдяки розробці методу стиснення без втрат інформації, що дозволяє зменшити розмір даних, не втрачаючи цілісності інформації.

Людські мови можуть бути представлені за допомогою ентропії, оскільки це також інформація, яка містить надлишковість. У контексті комп'ютерів будь-яка інформація кодується у двійковому вигляді «0» та «1», у бітах, і ентропія також виражається у бітах, відображаючи ступінь невизначеності в даному наборі даних.

Для визначення теоретичної межі стиснення даних без втрат використовується складність Колмогорова – найкоротший запис виводу вхідного повідомлення.

Це показує, скільки інформації потрібно для того, щоб визначити об'єкт. Іншим підходом є використання найкоротшої програми, яка надрукує певний рядок і зупиниться. Однак ця складність є необчислювальною функцією $x \rightarrow K(x)$, що означає, що немає алгоритму для обчислення цієї функції узагальнено.

$$K(x) := \min l(p) + l(y) \quad (9)$$

де, p – програма, y – вхідна інформація, x – вихідна, $l(p)$ – довжина вхідного p , $l(y)$ – довжина програми y .

Використовується функція $p(y) = x$, якщо програма може зупинитися після опрацювання всіх вхідних даних. Довжини визначаються кількістю бітів, що є показником на обсяг інформації, яку потрібно передати або зберегти.

Для ефективної роботи цього підходу пам'ять комп'ютера має бути необмеженою, щоб забезпечити достатній обсяг для зберігання всіх вхідних даних та іншої необхідної інформації.

Проблема зупинки виникає при спробі описати програму, яка приймає дані на вході і може або завершити виконання, або працювати нескінченно довго. Наприклад, в ситуації, коли потрібно закрити двері останній людині в аудиторії, важко визначити момент завершення дії.

Колмогоровська складність, представлена MDL, дозволяє стискувати дані, використовуючи принцип Оккама - не використовувати зайвих припущень. Будь-яка закономірність даних може служити для ефективного стиснення, використовуючи найменшу кількість інформації.

У теорії інформації, визначеної Шенноном, викладені принципи кодування, які спрямовані на забезпечення мінімальної середньої кількості кодових слів у джерелі інформації без шумів.

Згідно з цими принципами, ключовими вимогами є рівні ймовірності появи символів та незалежність ймовірностей від попередніх символів. Ці умови гарантують ефективність кодування та передачі інформації, забезпечуючи оптимальні умови для опрацювання інформаційного потоку без додаткових спотворень.

2.2. Принцип роботи Хаффмана

Сучасні методи стиснення даних використовують два основних підходи ентропійного кодування: методи Хаффмана та арифметичне кодування.

Арифметичне кодування, що входить до складу ентропійного підходу, працює за принципом кодування символів повідомлення у єдине раціональне число. Цей метод відрізняється від Хаффмана, оскільки для кожного символу він використовує лише одне число, а не свій власний набір бітів. Такий підхід дозволяє кодувати повідомлення з ентропією менше 1 біта на символ.

Щодо алгоритму Хаффмана, відомого своєю жадібністю, він визначає оптимальні коди для символів, враховуючи їхню частоту входження у повідомленні. Цей метод забезпечує мінімальну кількість бітів для стиснення повідомлення, враховуючи тільки символи, які фактично зустрічаються у вхідних даних.

Жадібний алгоритм, вперше винайдений Девідом Хаффманом у 1950-х роках, є методом, що на кожному кроці обирає найкращий варіант рішення, незалежно від подальших наслідків, вірячи, що такий підхід призведе до оптимального результату в цілому.

Основна ідея жадібного алгоритму Хаффмана полягає в використанні схеми кодування змінної довжини в залежності від частоти появи символів у повідомленні. Символи, які зустрічаються найчастіше, кодуються бітовим представленням, яке займає найменше місця, в той час як рідкі символи використовують більше біт. Для розкодування стиснутого повідомлення отримувачу потрібно передати метод декодування, а саме дерево Хаффмана, яке вказує, як розкодувати біти у символи ASCII(рис. 2.2).

При кодуванні зручно використовувати таблицю Хаффмана, що генерується після побудови дерева. Ця таблиця дозволяє ефективно перетворювати двійкові коди у символи та спрощує процес зчитування стиснутого файлу.

Таблиця 2.2 — кодування символів ASCII

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(72	48	110	H	104	68	150	h
9	9	11		41	29	51)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

Алгоритм Хаффмана гарантує повну декодованість, оскільки жодне закодоване слово не є префіксом іншого у створеному дереві. Наприклад, якщо ми маємо кодові слова 10 і 101, то одне не може бути префіксом іншого, що унеможлиблює конфлікти при декодуванні.

Під час початкового кодування відправляється створене дерево (header), яке визначає структуру кодових слів. Кодові слова представляють собою рядки бітів, які використовуються для закодування конкретних символів. Дерево Хаффмана, що виглядає як бінарне дерево, має n листових вузлів та $n - 1$ внутрішній вузол, де n - кількість унікальних символів у повідомленні (рис. 2.3).

Прохід по дереву Хаффмана здійснюється від кореня до його листків, що означає, що спочатку обробляються бітові представлення високочастотних символів. Для побудови дерева використовується пріоритетна черга або сортування за частотою зустрічі символів у повідомленні. Внутрішні вузли мають посилання на дочірні вузли "left" та "right" та містять вагу або частоту.

Використання умовних бітів «0» та «1» вказує на лівий та правий дочірній вузол відповідно. Такий метод кодування забезпечує оптимальність для будь-якого вхідного повідомлення.

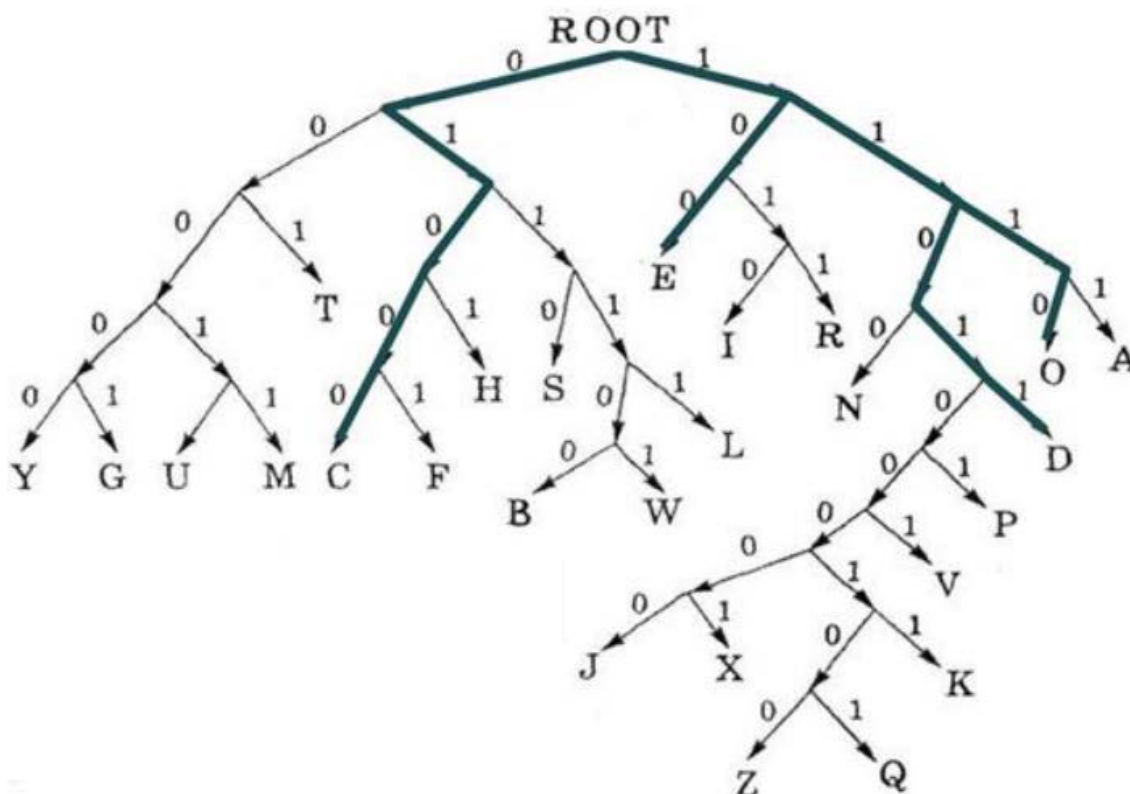


Рисунок 2.3 презентує Дерево Хаффмана, створене для англійської мови

З використанням цього дерева ми здійсимо кодування слова "Code", результати якого представлені в Таблиці 2.1.

Таблиця 2.1 відображає отримані коди, виведені із Дерева Хаффмана.

Символ	Кодування
C	01000
O	1110
D	11011
E	100

Код Хаффмана досягає максимальної ефективності, коли ймовірності виникнення подій є негативними ступенями двійки. Це через те, що формула

для обчислення кількості бітів є двійковим логарифмом, який повинен бути цілим числом, що забезпечує найкращу можливу стискальну ефективність.

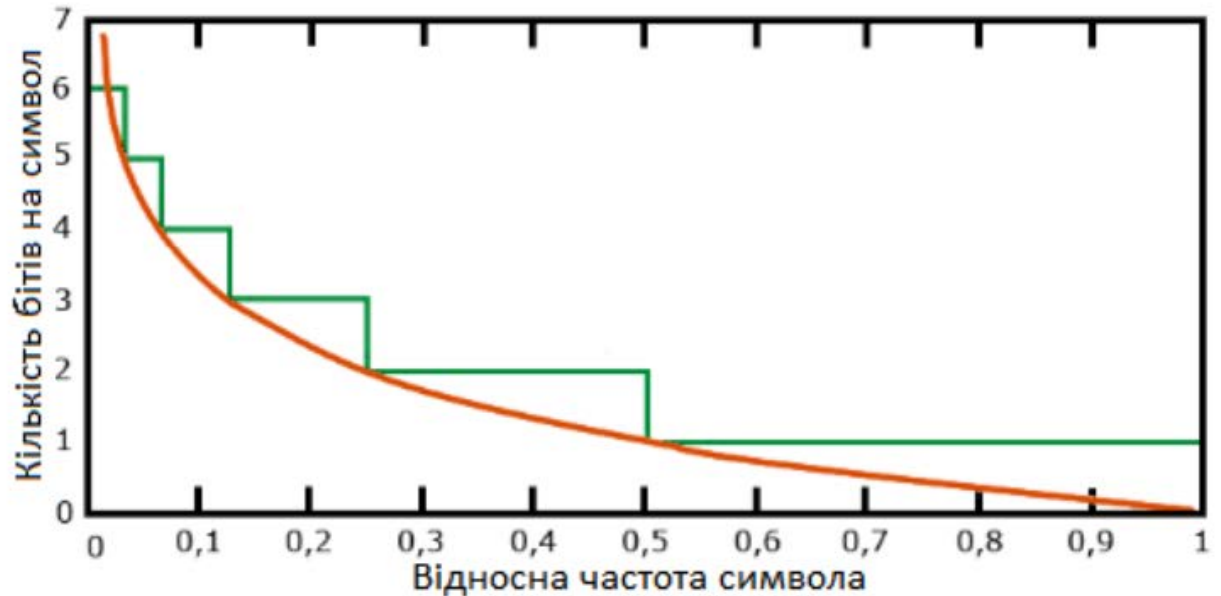


Рисунок 2.4 – Порівняння Хаффмана з оптимальним стисненням

Метод Хаффмана виявляється значно швидшим у порівнянні з арифметичними чи інтервальними методами, але найефективніший він при апроксимації для ступенів двійки. Коли відносні частоти не є ступенями двійки, ефективність алгоритму може зменшитися. Крім того, алгоритм погано справляється з ймовірностями, які не є цілими числами, наприклад, $1/3$. Оптимальна кількість бітів для цього символу – 1,6, але Хаффман, здебільшого, видає цілі числа, що може призвести до втрати ефективності стиснення.

Інші алгоритми можуть вимагати значних обчислювальних витрат для того, щоб наблизитися до теоретичної ентропії Шеннона. Для апроксимації цього процесу може використовуватися метод Монте-Карло, який моделює відтворення приблизних явищ.

Питання оптимальності побудови такого дерева є важливим, оскільки використання найчастіше зустрічається символу у вхідному повідомленні дозволяє мінімізувати проходження по дереву, що визнається як оптимальний підхід.

Для опису двійкового коду змінної довжини слід зазначити умови, які передбачають:

- для заданих слів a_j , a_k , якщо $P[a_j] \geq P[a_k]$, то $l_k \geq l_j$, де l – кількість біт для заданого слова a_j ;
- дві літери з найменшими ймовірностями мають кодові слова з однаковою кількістю біт l_m ;
- дерево має мати дві гілки, що відходять від кожного проміжного вузла;
- Якщо початкове дерево є оптимальним для визначеного алфавіту, то зменшене дерево повинно залишатися оптимальним для зменшеного алфавіту. У випадку, якщо це не виконується, необхідно виявити код з меншою середньою довжиною, забезпечуючи оптимальні умови для нового алфавіту.

Для визначення ефективності алгоритмів застосовують обчислювальну складність, яка визначає кількість часу та/або пам'яті, необхідної алгоритму для обробки вхідних даних при певному їх розмірі. Поняття Big-O використовується для вказівки на верхній межі часової складності алгоритму, тобто його найгіршого можливого сценарію, коли програма працює найтриваліше.

Найгірший сценарій або алгоритмічна складність Хаффмана зазвичай виражається як $O(n \log(n))$, де n - кількість символів у вхідному алфавіті. У випадку, коли символи відсортовані за частотою, складність алгоритму буде лінійною, що вказує на його високу ефективність у таких умовах.

Розглянемо приклад побудови дерева Хаффмана:

- а) На вхід подаються символи з вказаною кількістю їх з'явлень у повідомленні.
- б) Для символів з найменшими частотами будується піддерево.
- в) Формується піддерево з наступних символів.
- г) Піддерева об'єднуються.
- г) Додається останній символ з батьком – сумою всіх частот.

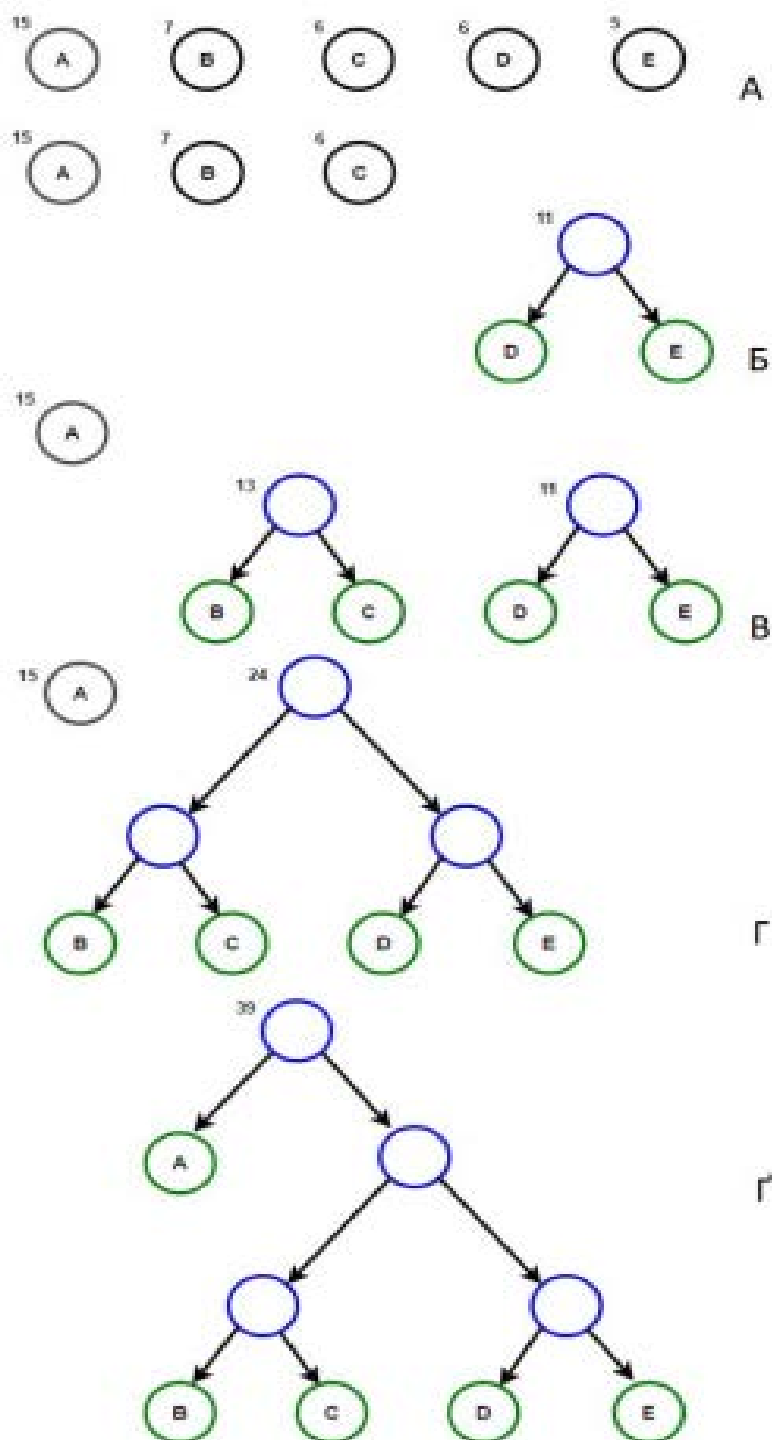


Рисунок 2.5 ілюструє процес створення кодового дерева:

Кожен символ у даній послідовності зараз займає 8 біт, що становить загалом 312 біт ($8 * 39$). Проте, використовуючи алгоритм Хаффмана, ми успішно стискаємо цю послідовність до 87 біт, досягнувши стиснення в 3.59 рази або приблизно 72%. Це означає, що в середньому ми витрачаємо 2.23 біти на кожен символ.

Процес конструювання самого дерева включає в себе обрання двох найменш часто зустрічаючих символів (D і E) і створення спільного листка, вага якого є сумою частот цих двох символів. Біти для представлення символів обираються з кінця до початку, і вони є оптимальними префіксами через мінімальні частоти символів.

Алгоритм Хаффмана демонструє низьке відхилення ентропії при великих алфавітах. Серед переваг кодування Хаффмана можна виділити відсутність префіксів у двійкових кодах, змінну довжину кодів та короткі коди для часто зустрічаючих символів.

Однак існують недоліки, такі як двоетапний процес проходження по дереву, обмеженість застосування тільки для текстової інформації, важкість реалізації для великих алфавітів, можливість втрати даних при зберіганні дійсних чисел та проблеми з перевіркою пошкоджень даних.

Під час декомпресії стисненого повідомлення спочатку будується дерево Хаффмана на основі заголовка, який був переданий на початковому кодуванні. Після цього, під час зчитування бітів, застосовуються правила, наведені в заголовку: 0 вказує на лівий вузол, 1 – на правий. Процес декомпресії завершується, коли досягається кінець вузла, а утворена послідовність бітів перетворюється на символ ASCII. Декомпресія завершується, кількість розкодованих символів дорівнює третє число, яке відповідає кількості символів у стислому файлі.

Існує адаптивний алгоритм Хаффмана, такий як алгоритм Фоллера-Галлаера-Кнута (FGK) або його інша реалізація, наприклад, алгоритм Віттера. Ці алгоритми дозволяють уникнути передачі статичних даних, адаптуючись до зміни характеристик джерела. Наприклад, адаптивний алгоритм Хаффмана може динамічно будувати дерево під час читання вхідних даних, що робить його більш гнучким, але при цьому можна працювати трошки більше.

У порівняльному аналізі на (рис. 2.6) використання, що адаптивний алгоритм Хаффмана вимагає більше часу, після чого він побудований

динамічно під час читання вхідних даних. Однак цей метод робить лише один прохід під час отримання повідомлення, що може бути

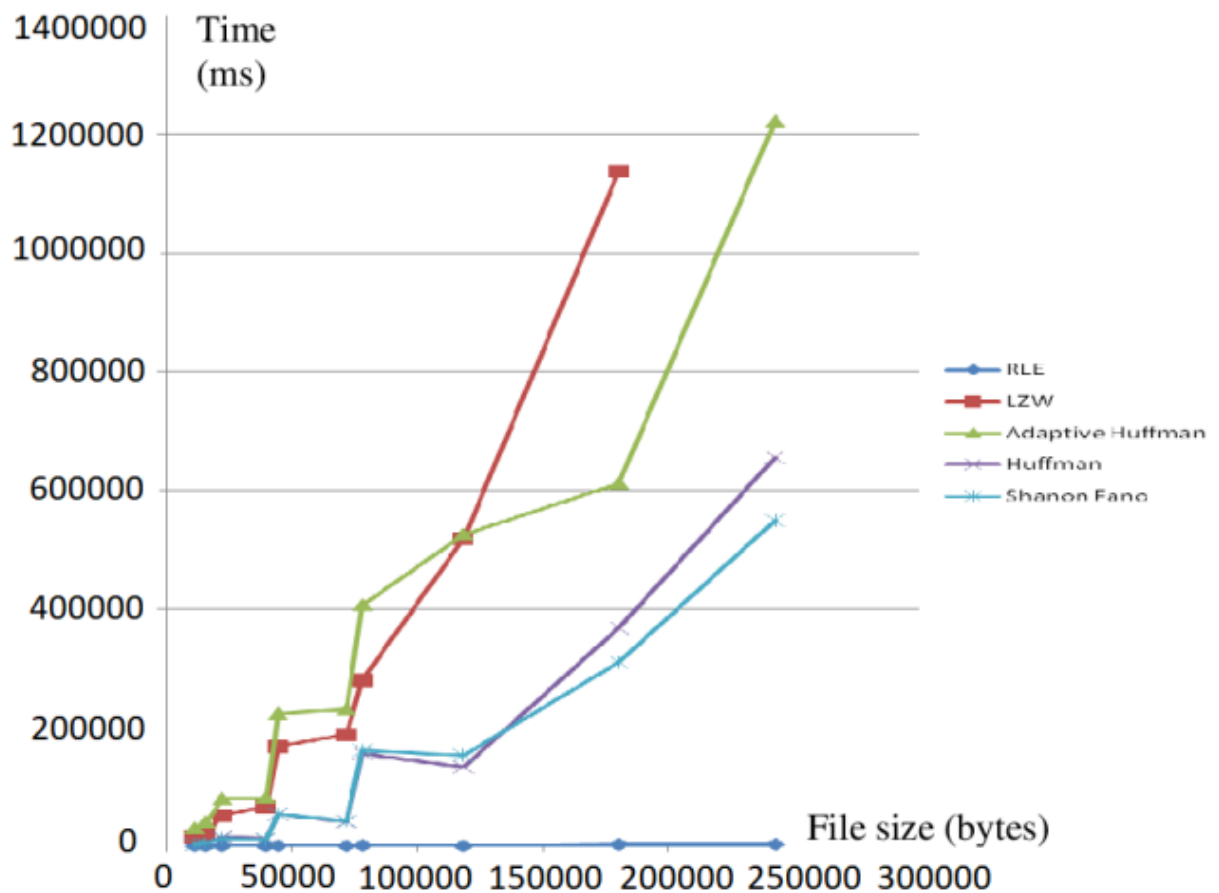


Рисунок 2.6 – Порівняння часу до вихідного розміру файлу після стиснення у різних алгоритмах

Дерева оновлюються зі змінною характеристикою на шкірній стороні відповідно до контексту: під час кодування це оновлення відбувається на етапі передачі, а при декодуванні – під час процесу приймання.

Цей метод актуалізації структури Хаффмана використовується у випадку, коли неможливо отримати таблицю частот для символів, особливо при передачі даних через мережу, де умови можуть змінюватися.

За допомогою адаптивного оновлення дерев Хаффмана система може ефективно застосовуватися до змін у потоках даних, що є критичним в умовах, коли передавач та приймач не мають інформації про частоту символів вперед.

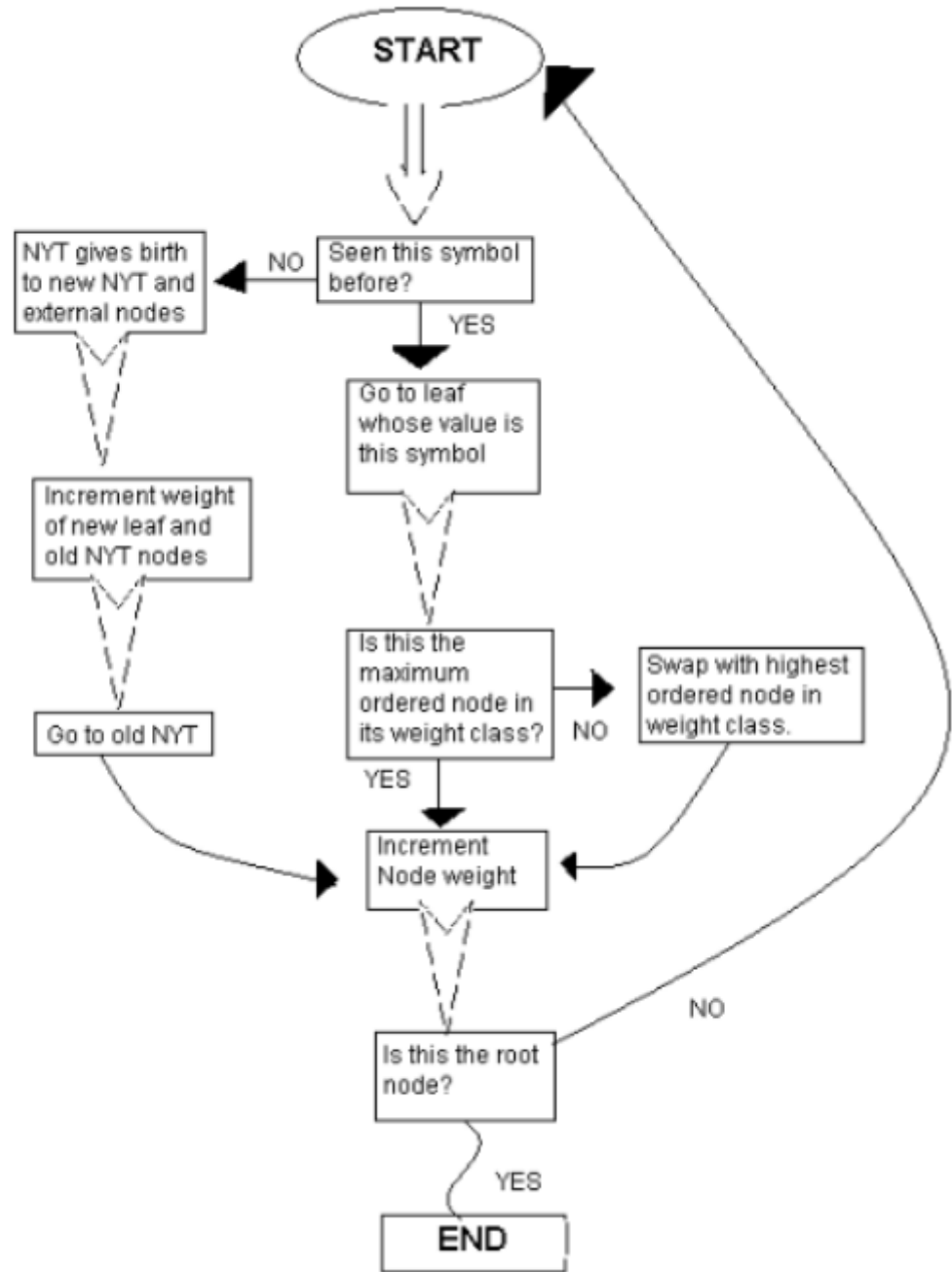


Рисунок 2.7 – Схема роботи адаптивного алгоритму Хаффмана

Розглянемо приклад побудови адаптивного дерева (рис. 2.7):

У модифікації буде застосовано метод стиснення представлення дерева, що дозволить більш ефективно зберігати пам'ять та покращить ступінь компресії. Цей підхід передбачає використання спеціально стиснутого представлення для дерева під час процесу кодування. Під час декодування використовується вміст header, який включає в себе саме це стиснене

1. Загальна кількість символів у вихідному повідомленні, що дозволяє зберігати контекст оригінального тексту.
2. Загальна кількість символів у стисненому вигляді, вказуючи на результат ефективності стиснення.
3. Топологія дерева кодування Хаффмана, яка визначає структуру та ієрархію використаних кодів.

Хаффман в цілому вирішував завдання стиснення, оптимізуючи використання байтів, де кожна одиниця інформації представлена окремим байтом, що містить 8 бітів. Це підходить для роботи з текстовою інформацією, де кожен символ може бути закодований відповідним байтом, забезпечуючи ефективний обмін і збереження даних (рис. 2.8).

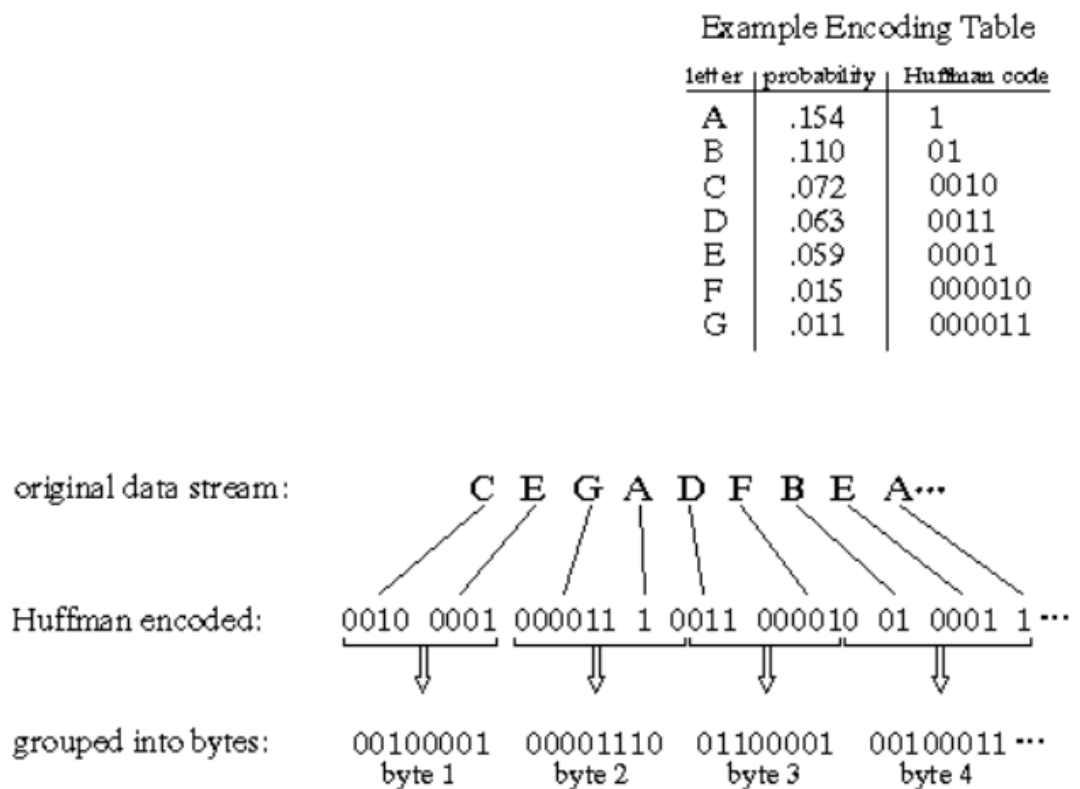


Рисунок 2.9 – Приклад зчитування даних в байтах відносно бітових значень.

Але проблемою Хаффмана є наявність невикористаних бітів, які не несуть корисної інформації, утворюючи зайвий обсяг даних. Для представлення англійської мови ми використовуємо ASCII – систему кодів, де

кожен символ може бути представлений трьома видами кодування: десятковим, шістнадцятковим та двійковим. Це дозволяє забезпечити універсальний метод кодування для текстової інформації, але виникає питання ефективного використання бітів у контексті стиснення даних.

Ми використовуємо алгоритм Хаффмана для стиснення та подальшого порівняння кількості бітів, які витрачаються під час компресії, з двійковим представленням символів у таблиці ASCII. У звичайному текстовому файлі, де кожен символ кодується 8 бітами за стандартом ASCII, або навіть 16 бітами за Unicode, ефективність стиснення може бути порівняна.

Важливо відзначити, що ASCII-таблиця включає всього 128 символів (від 0 до 127), при цьому один верхній біт залишається невикористаним. Заповнення цього біта при поверненні до батьківського листка дозволяє вказати положення дочірнього листка (0, якщо дочірній листок знаходиться ліворуч від батьківського, 1 – праворуч).

Для n елементів кількість ребер у дереві буде $(2n - 2)$, але на кожен символ та ребро в оригінальному Хаффмані потрібно по 8 біт, що у сумі дасть загальну кількість бітів $(16n - 2)$.

Теоретично, використовуючи запропонований метод, ми можемо очікувати економію $(7n + 2)$ бітів у порівнянні з оригінальним Хаффманом. Проте, важливо відзначити, що наша модифікація має певні недоліки, основним з яких є складніша реалізація декомпресії. Це може вплинути на швидкість декодування та загальну ефективність алгоритму в реальних умовах використання.

2.3 Кодування Хаффманом наборів символів

У попередніх розділах ми аналізували алгоритм в контексті окремих символів; однак його можна успішно використовувати для наборів чи навіть цілих слів. Важливо відзначити, що в повсякденному словнику людей часто використовується обмежена кількість слів, яка, за деякими оцінками, не перевищує дві тисячі.

Метод відрізняється тим, що, в відміну від звичайного алгоритму, який використовує лише вхідне повідомлення для побудови таблиці ймовірностей, у нашому випадку ми враховуємо ймовірність появи слів чи наборів символів, враховуючи особливості самої мови.

$$P(a) \log_2 P(a) + P(b) \log_2 P(b) + P(c) \log_2 P(c) \leq P(ab) \log_2 P(ab) + P(ac) \log_2 P(ac) + P(bc) \log_2 P(bc) \quad (10)$$

де, a, b, c – токени.

Для оцінки можна розглянути приклад:

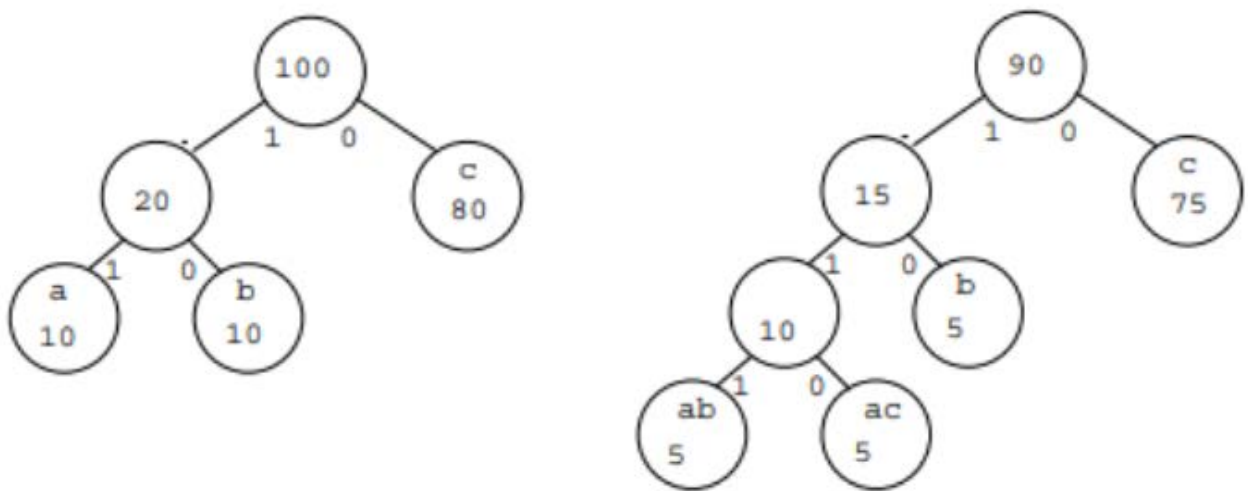


Рисунок 2.10 презентує Дерева Хаффмана, спроектовані для наборів токенів.

З лівого дерева видно, що токени 'a' повторюються 10 разів, 'b' – 10, 'c' – 80. Розглянемо ситуацію, де у половині випадків після літери 'a' йдуть 'b' та 'c', що дозволяє побудувати праве дерево. Ліве дерево складає 120 біт, тоді як праве – 115, зберігаючи 5 бітів через цю агрегацію. Проте важливо враховувати, що стиснення у таких випадках не завжди є можливим, і застосування агрегації вимагає обережності.

2.4. Застосування алгоритму Хаффмана

Основне використання алгоритму полягає в обробці текстів, оскільки це алгоритм без втрат даних, що важливо для текстових файлів. Однак існують інші області застосування, які ми розглянемо. Наприклад, алгоритм може бути

застосований у монохромних зображеннях (рис. 2.11), де кожен піксель приймає значення від 0 до 255 (рис. 2.12).

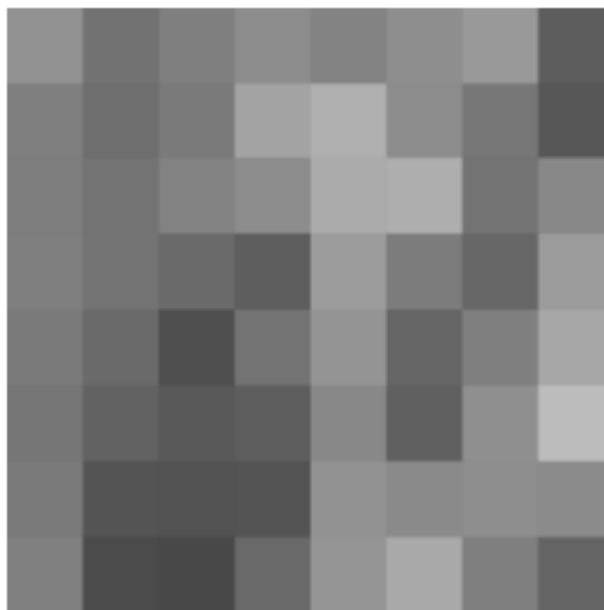


Рисунок 2.11 – Монохромне зображення

128	75	72	105	149	169	127	100
122	84	83	84	146	138	142	139
118	98	89	94	136	96	143	188
122	106	79	115	148	102	127	167
127	115	106	94	155	124	103	155
125	115	130	140	170	174	115	136
127	110	122	163	175	140	119	87
146	114	127	140	131	142	153	93

Рисунок 2.12 – Представлення зображення у вигляді числових значень

Хаффмана

Масив інтерпретованої послідовності містить 46 унікальних значень інтенсивностей, що будуть закодовані у 46 різних послідовностях бітів.

Розглянемо приклад на (рис. 2.13):

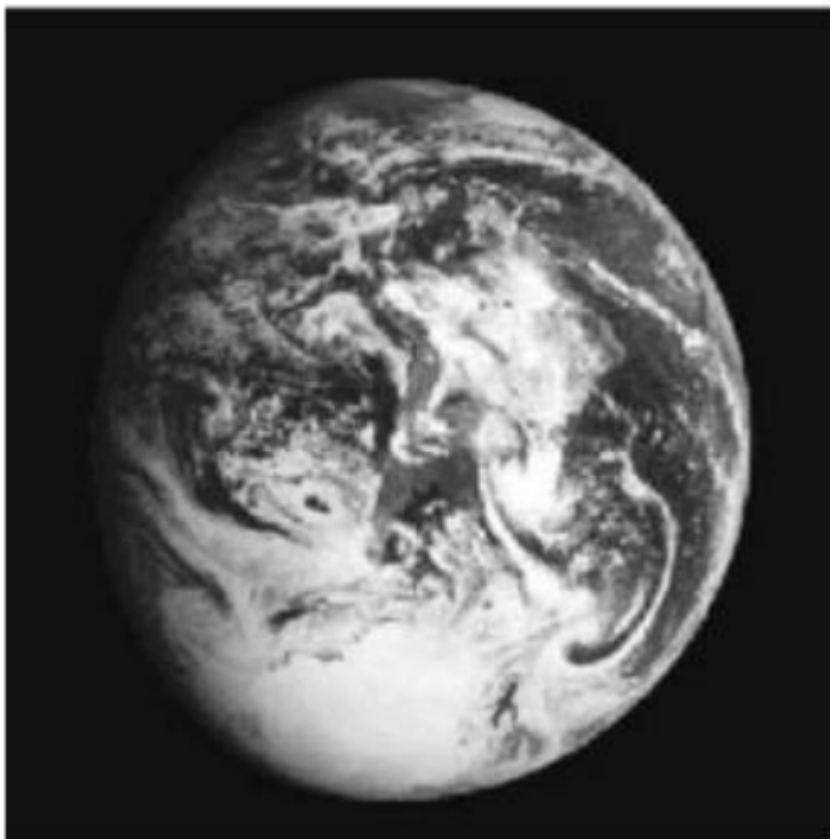


Рисунок 2.13 – Приклад стиснення монохромного зображення за алгоритмом Хаффмана

Загальний розмір зображення складає 40,534 байти, що вказує на витрату 4,94 бітів на один піксель. При цьому величина коефіцієнта стиснення становить 1,62. Детальний аналіз показує, що для деяких пікселів характерно однакове або дуже схоже оточення. Після стиснення витратили лише 4,13 бітів на піксель, при цьому обсяг файлу зменшився до 33,880 байтів, а коефіцієнт стиснення досяг 1,93.

Використовуючи адаптивний алгоритм Хаффмана, ми отримали наступні результати: обсяг файлу – 39,504 байти, витрати 4,82 бітів на піксель, а коефіцієнт стиснення склав 1,66. Навіть якщо оригінальний алгоритм показав кращі результати, перевагою адаптивного методу є його здатність до онлайн застосування. Важливо відзначити, що адаптивний метод, будучи більш динамічним, може бути вразливішим до помилок, але його реалізація залишається більш складною. Також, алгоритм Хаффмана широко використовується у великих корпораціях, таких як Google. Ці компанії застосовують алгоритми стиснення без втрат, які базуються на комбінації

методів Хаффмана та LZ77. Недавно, Google випустила новий алгоритм, Brotli (рис. 2.14), який продемонстрував 20-відсоткове покращення ефективності в порівнянні з попередніми методами стиснення.

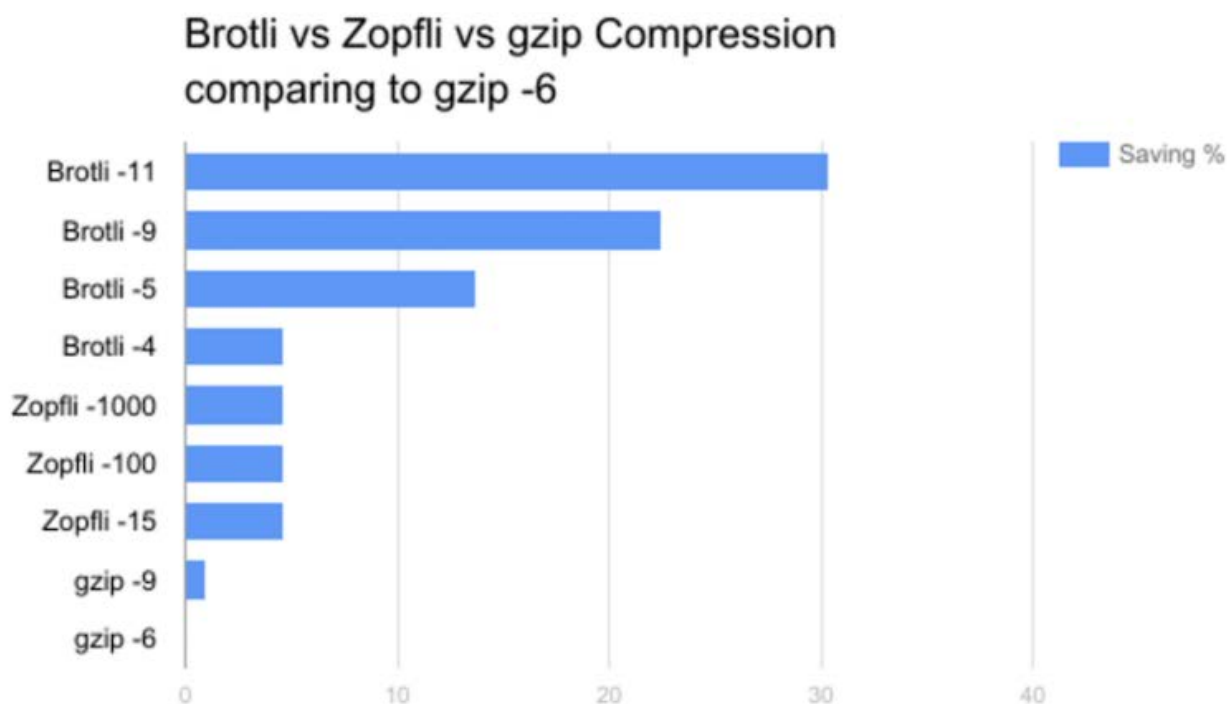


Рисунок 2.14 – Порівняння алгоритмів стиснення

У 2014 році з'явився інноваційний алгоритм ANS, який наближається до ентропійно оптимального кодування та відзначається високою швидкістю. На сьогоднішній день цей алгоритм не поступається швидкістю Хаффмана. В даному методі використовується ентропійне кодування для визначення наступного символу. ANS широко застосовується у ZSTD (Facebook), який комбінує обидва зазначені алгоритми.

Також алгоритм ANS використовується в архіваторах, таких як WINZIP, GZIP (рис. 2.15), PKZIP, BZIP2. Префіксні коди з цього методу застосовуються в стандартах JPEG (рис. 2.16), PNG, MP3, а також в технології факсимільного передавання.

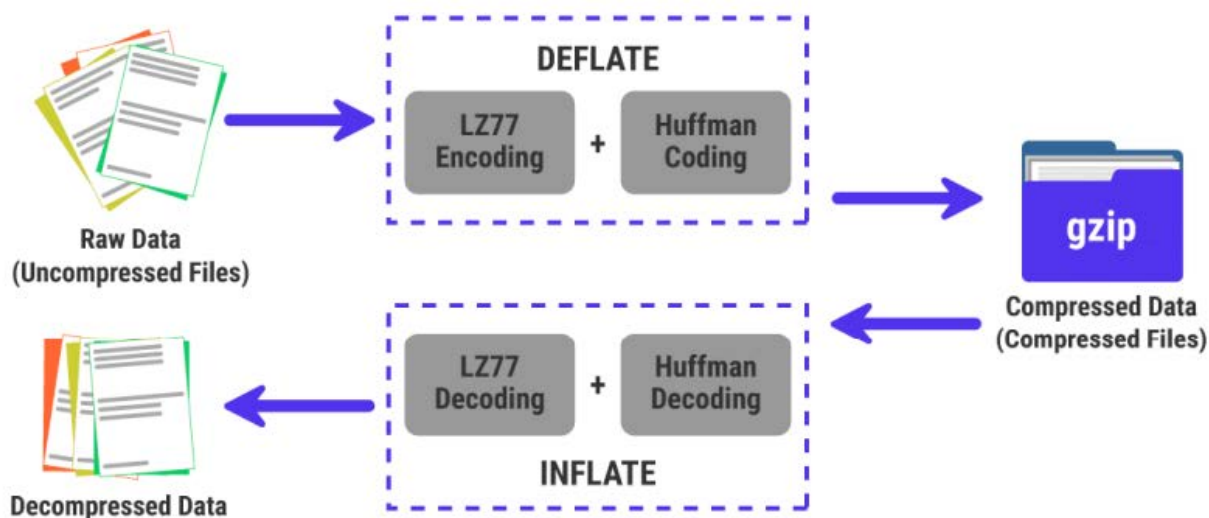


Рисунок 2.15 – Приклад застосування в архіваторі GZIP

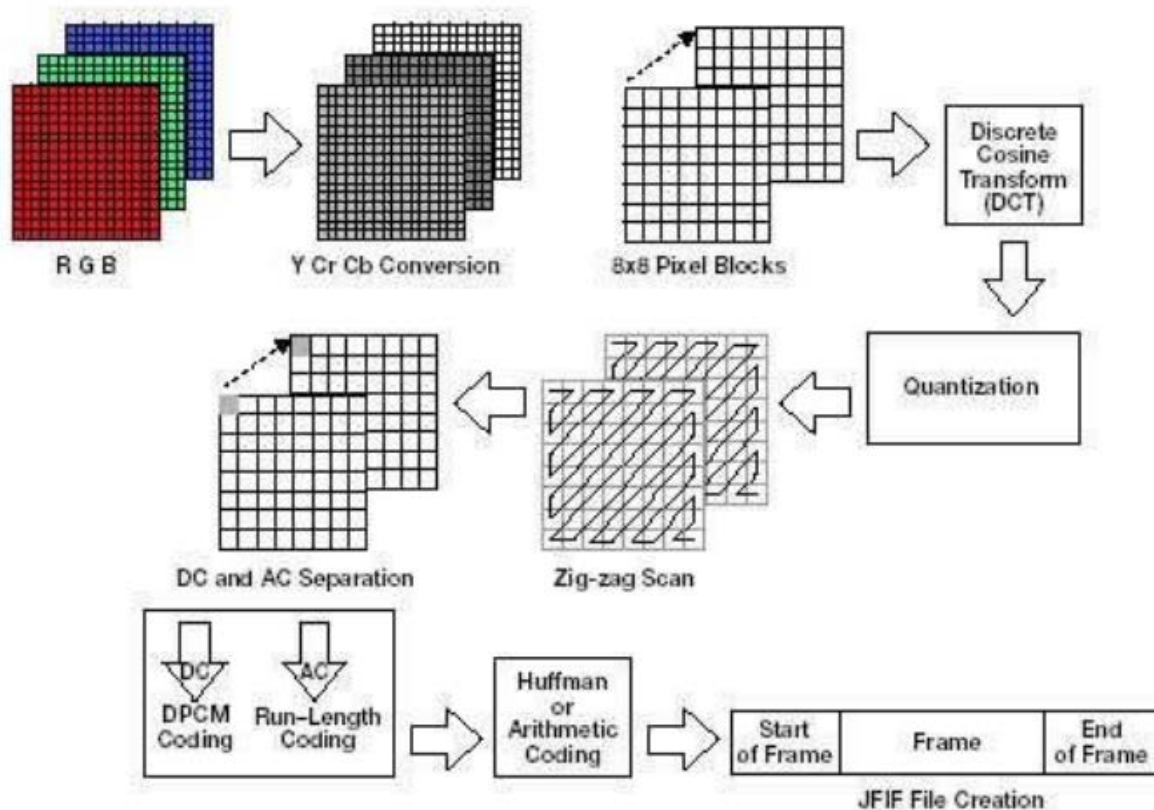


Рисунок 2.16 – Приклад застосування алгоритму Хаффмана в кодуванні базового потоку JPEG

Наприклад, GZIP використовує як основний алгоритм LZ77 та два додаткові алгоритми Хаффмана. Перший алгоритм визначає послідовність рядків, а другий використовується для стиснення відстані, якщо рядок вже був попередньо стиснутий.

Це ще раз підтверджує активність розробки нових методів стиснення даних, спрямованих на ефективне використання ресурсів хмарових технологій. Ці методи дозволяють не лише зменшити трафік Інтернету, а й прискорюють завантаження веб-сторінок, що, в свою чергу, сприяє економії заряду батареї пристроїв.

2.5 Оцінка стиснення даних

При розробці подібних алгоритмів важливо одночасно враховувати кілька факторів: швидкість роботи, складність реалізації (з урахуванням використання пам'яті та ресурсів постійно запам'ятовуючих пристроїв), обсяг оперативної пам'яті, формати стиснених файлів і сумісність з основними архіваторами. Правильне балансування цих аспектів дозволяє створити ефективний алгоритм стиснення, який задовольнить потреби користувачів у різних сценаріях використання.

Оцінки швидкості:

1. Декомпресія, що представляє собою процес відновлення стиснутих даних у їхню первинну форму, визначається як кількість стиснутих бітів до часу, який витрачений на декомпресію. У деяких сценаріях, таких як в навушниках, де стандартна швидкість стереозвуку не підтримується, це може мати критичне значення і ускладнити їхнє ефективне використання.
2. Стиснення визначає відношення між нестиснутими бітами та часом, витраченим на компресію. Наприклад, якщо аудіокомпресор не підтримує швидкість 56 Кбіт/с, програмне забезпечення може створювати безшумні інтервали, що потенційно впливає на якість звуку.

Ці швидкості залежать від різноманітних чинників, таких як характеристики комп'ютера, використання конкретного компілятора, розмір файлу та його формат. Зазначено, що швидкість стиснення, як правило, залишається на рівні, нижчому за швидкість декомпресії.

Для покращення ефективності на різних вхідних послідовностях повідомлень можна використовувати перетворення Берроуза-Вілера. Це оборотне перетворення, яке не потребує додаткового зберігання інформації. Хоча воно не є алгоритмом стиснення в справжньому розумінні, але перетворює вхідне повідомлення так, щоб його було зручно стискувати. Знаходження послідовності здійснюється циклічними ротаціями на всю довжину n , лексикографічно сортуючи результати. Цей підхід, який спочатку використовує масив рядків для зберігання всіх можливих варіантів вхідного повідомлення, особливо ефективний при обробці великих обсягів інформації.

Цей метод застосовується для стиснення без втрат, як у випадку алгоритму bzip2. Крім того, проводиться оцінка ступеня стиснення та потужності.

$$k = \frac{s_0}{s_c} \quad (10)$$

де k – коефіцієнт стиснення, s_0 – об'єм вхідних даних, s_c – об'єм стиснених.

Усі алгоритми стиснення даних дають різні коефіцієнти стиснення даних для різних файлів (рис. 2.17).

Для оцінки ефективності збереження простору використовується відношення обсягу вихідних даних до обсягу вхідних даних, з якого віднімається одиниця. Важливо відзначити, що через внутрішню ентропію даних алгоритми стиснення без втрат, як правило, досягають відношення стиснення, яке не перевищує 2:1.

Крім того, можна розглядати стиснення як особливий випадок розходження даних, оскільки процес кодування та декодування часто здійснюється на різних пристроях.

Запропонований метод включає в себе скремблювання тексту, перестановку, перетворення Берроуза-Вілера (BWT), переміщення вперед (MTF) і кодування довжини пробігу для підвищення стійкості шифрування і досягнення ефективного стиснення даних. Таке поєднання методів пропонує

потенціал для ефективного стиснення, забезпечуючи при цьому надійні засоби шифрування.

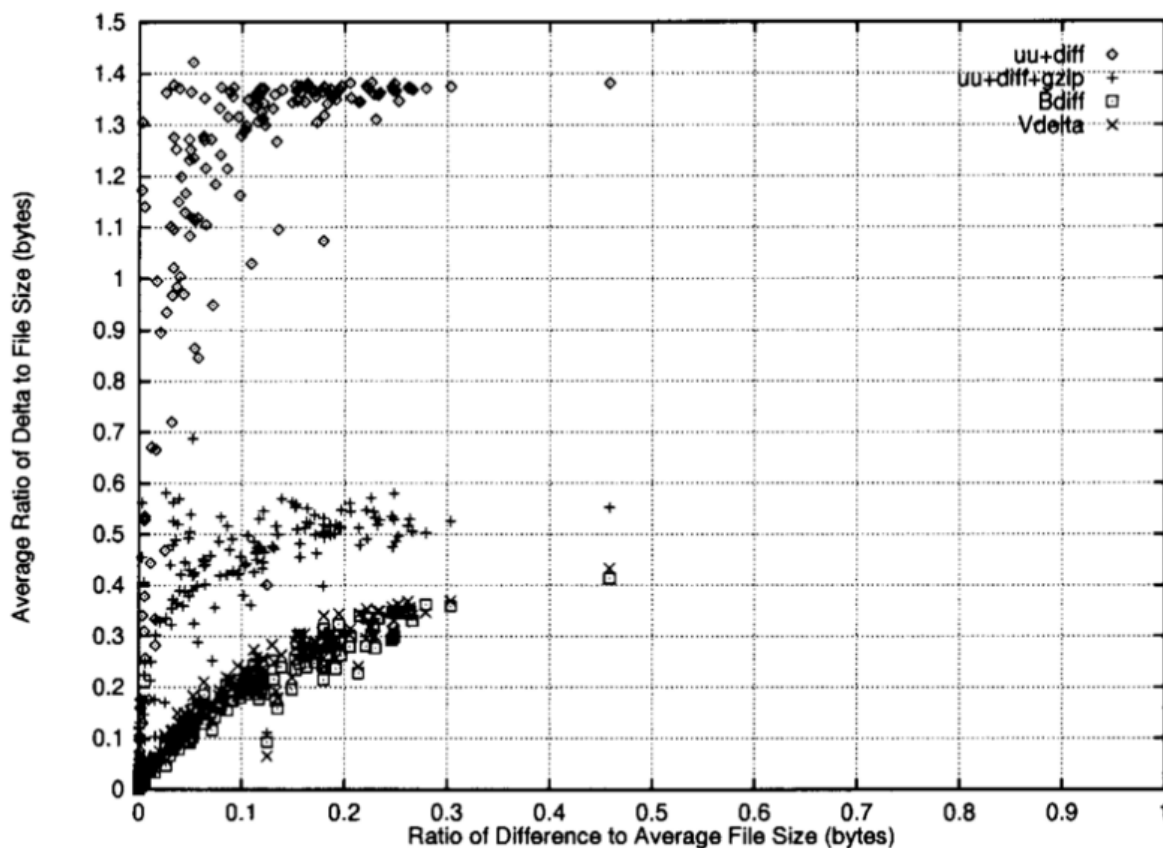


Рисунок 2.17 – Коефіцієнт стиснення в порівнянні з розмірами вхідних даних та використовуваними алгоритмами.

У поєднанні ці методи створюють надійну схему стиснення. Спочатку текст зашифровується, щоб внести випадковість і порушити закономірності. Потім послідовно застосовуються перестановка, BWT, MTF і кодування довжини циклу для подальшого зменшення розміру даних. Ця комбінація дозволяє ефективно стискати дані, зберігаючи при цьому їх цілісність і конфіденційність.

Поєднання цих методів гарантує, що зашифровані дані залишаються захищеними завдяки надійним механізмам шифрування, водночас досягаючи достатнього стиснення за рахунок використання шаблонів і надлишковості в даних. Інтеграція методів шифрування і стиснення в узгоджений спосіб дозволяє досягти цілей безпеки та ефективності.

Скремблювання тексту - це перестановка символів у тексті, щоб ускладнити його розшифрування. Ця техніка може внести випадковість і порушити шаблони, підвищуючи безпеку даних.

Перестановка передбачає зміну порядку елементів у послідовності. Позиції символів змінюються шляхом застосування пермутації до тексту, що ще більше заплутує початкове повідомлення.

Перетворення Берроуза-Вілера (BWT) - це оборотна техніка перетворення даних, яка переставляє символи в залежності від контексту. Він групує схожі символи, роблячи подальші алгоритми стиснення більш ефективними.

Move-to-Front (MTF) - простий, але ефективний метод, який реорганізує набір символів на основі частоти їх появи. Символи, що часто зустрічаються, розміщуються попереду, що забезпечує ефективне кодування під час стиснення.

Кодування довжини прогону - це метод стиснення, який замінює послідовно повторювані символи або послідовності на підрахунок і одиничне входження. Це значно зменшує розмір даних, що повторюються.

Запропонована робота підвищує безпеку даних за допомогою комбінованого підходу до шифрування та стиснення, не впливаючи на продуктивність один одного. Підхід до безпечного та ефективного стиснення заснований на використанні скремблорованого входу. Це досягається шляхом генерації ключа і застосування перестановки. Як значення ключа використовується частота символів у вхідному тексті. Запропонований метод включає в себе скремблювання вхідних даних за допомогою згенерованого ключа, перетворення їх за допомогою BWT та стиснення за допомогою кодування Move-To-Front та Run-Length Encoding. Процес стиснення інтегрує властивості криптографії плутанини та дифузії, покращуючи продуктивність за рахунок вимірювання відстані унікальності параметра. Блок-схема кодера в запропонованій системі показана на рис. 2, який також ілюструє процес роботи запропонованої системи.

Запропонована система складається з чотирьох кроків:

1. Двоетапний процес скремблювання підвищує безпеку вхідної послідовності перед застосуванням перетворення Берроуза-Вілера. На першому етапі генерується унікальне ключове значення для кожного входу однакового розміру, яке визначається розподілом частот літер. Це ключове значення потім використовується для перетворення вхідних даних у форму $matriX$, а на другому кроці до $matriX$ застосовується перестановка стовпців. Після завершення перестановки дані знову перетворюються у векторну форму для подальшої обробки. Для визначення частотного розподілу символів в алфавітному порядку необхідно обчислити ключ. Цей ключ може бути піднесений до квадрату, коли довжина тексту має цілий квадратний корінь, що дорівнює розміру ключа. Таке скремблювання має на меті зашифрувати вхідні дані, зробивши їх більш безпечними для подальшої обробки.

2. Після того, як вхідна послідовність зашифрована, вона обробляється за допомогою перетворення Берроуза-Вілера (BWT) для кращого стиснення. Алгоритм сортування блоків BWT сортує блоки даних для покращення стиснення. Алгоритм починається з виконання N циклічних зсувів вправо на зашифрованому вході, після чого виконується лексикографічне сортування. З відсортованої матриці X отримується останній символ кожного повороту як результат кодування BWT. BWT - це алгоритм блокового сортування без втрат, який сортує блоки даних без їх стиснення. Вхідними даними для процесу кодування за допомогою BWT є зашифрований файл.

3. Вихідні дані з BWT передаються на кодер Move-To-Front Transform. Це перетворення зменшує ентропію даних, але не призводить до стиснення. Перетворення Move-To-Front (MTF) - це метод стиснення даних, який змінює розташування символів у вхідній послідовності так, щоб символи, які найчастіше зустрічаються, з'являлися раніше. Цей підхід передбачає присвоєння кожному символу у вхідних даних певного місця, і щоразу, коли символ зустрічається, він переміщується на початок списку. В результаті, коли

той самий символ з'являється знову, він, швидше за все, матиме нижчу позицію у списку, яка може бути закодована меншою кількістю бітів. Результатом такого кодування є послідовність цілих чисел, яка вказує на позицію символів у списку.

4. Кодер Run-Length виконує завершальний етап процесу кодування, що дозволяє зменшити розмір послідовності даних. MTF часто використовується з іншими методами кодування, такими як Run-length кодування, для досягнення кращого стиснення. Цей простий і ефективний метод легко реалізується і може бути інтегрований в систему стиснення. RLE - це схема кодування, в якій послідовні повторювані значення замінюються одним входженням і підрахунком. Метою RLE є стиснення наборів, які містять довгі послідовності повторюваних значень, шляхом представлення їх у вигляді одного значення та кількості повторень.

Далі виконується декодування довжини відрізка, яке обертає процес кодування і відновлює вихідні дані, замінюючи послідовні повторювані значення одним входженням і підрахунком.

Після декодування довжини відрізка застосовується зворотне перетворення "зсув до початку". Це перетворення є зворотною операцією до перетворення Move-To-Front, що використовується під час кодування. Воно переставляє символи в послідовності на основі їхніх позицій у списку.

Результатом зворотного перетворення Move-To-Front є стовпці матриці перетворення Берроуза-Вілера (Burrows-Wheeler Transform, BWT). Цей крок допомагає відновити зашифровані вхідні дані.

Для відновлення початкових вхідних даних створюється векторна таблиця на основі стовпців матриці BWT. Ця векторна таблиця допомагає зіставити стовпці з їх початковими позиціями.

Нарешті, реконструйовані дані проходять операцію зворотної перестановки з використанням того ж значення ключа, яке спочатку використовувалося для скремблювання вхідних даних. Ця операція успішно дескремблює вхідні дані, відновлюючи вихідну послідовність.

Блок-схема та опис процесу декодування допомагають проілюструвати, як запропонована система змінює місцями кроки процесу кодування для отримання вихідних вхідних даних.

Для визначення рівня стиснення можна обчислити відношення стисненого вихідного розміру до вихідного нестисненого вхідного розміру. Кількісна оцінка інформаційного вмісту даних передбачає обчислення бітрейту. Для оцінки ефективності методу стиснення використовується декілька вимірів, зокрема бітрейт вихідних даних, ентропія вхідних і вихідних даних, відстань унікальності для різних розмірів ключів, а також коефіцієнт стиснення для різних розмірів вхідних даних. Після стиснення дані мають менше бітів на символ, а їхня ентропія більша, ніж у вихідної вхідної послідовності. Зі збільшенням розміру ключа відстань унікальності також зменшується, а ступінь стиснення може коливатися залежно від розміру вхідного файлу.

3 ОПИС РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАСТОСУНКУ

3.1 Засоби втілення програмного алгоритму стиснювання даних

Для втілення програми з порівняння модернізації алгоритму Хаффмана з його оригінальним представленням використано різноманітні інструменти для тестування, візуалізації та розробки. Зокрема, важливим етапом було вибір мови програмування C++, оскільки вона базується на низькорівневій мові C і має можливість ефективно працювати з бітами, що є необхідним для реалізації алгоритму.

Слід зазначити, що C++ відрізняється від мови C тим, що вона є об'єктно-орієнтованою та підтримує концепції об'єктів та класів. Розроблена Б'ярном Страуструпом, ця крос-платформова мова середнього рівня надає велику гнучкість у роботі з ресурсами та пам'яттю (рис. 3.1). Додатки, написані на C++, відзначаються високою продуктивністю, оскільки розробник може безпосередньо контролювати ресурси та пам'ять.

C++ також компілюється в набір машинних кодів цільової системи відразу перед виконанням, що сприяє ефективності програм, написаних на цій мові. Ця особливість особливо корисна у випадках, коли вимагається велика швидкодія.

Мова програмування C++ широко застосовується у написанні алгоритмів, де важлива швидкість взаємодії, наприклад, у розробці компіляторів, графічних програм та двигунів для ігор, де потрібен швидкий рендер. Ця мова також використовується в банківській сфері для обробки мільйонів транзакцій та інших областях застосування.

При розробці ігор та інших галузях, які активно використовують графіку, критично важливо дотримуватися оптимізацій для пропускну здатності графічного процесора (GPU). Оскільки центральний процесор та GPU є окремими фізичними пристроями, синхронізація між ними вимагає значних зусиль та часу. Таким чином, мінімізація операцій та обсягу

передаваних даних є важливим аспектом. Це може бути досягнуто за допомогою буферизації та оптимізації об'єму передаваних даних, включаючи їх стиснення. Використання мови програмування C++ та ефективних алгоритмів компресії інформації грає ключову роль у досягненні цих оптимізацій.

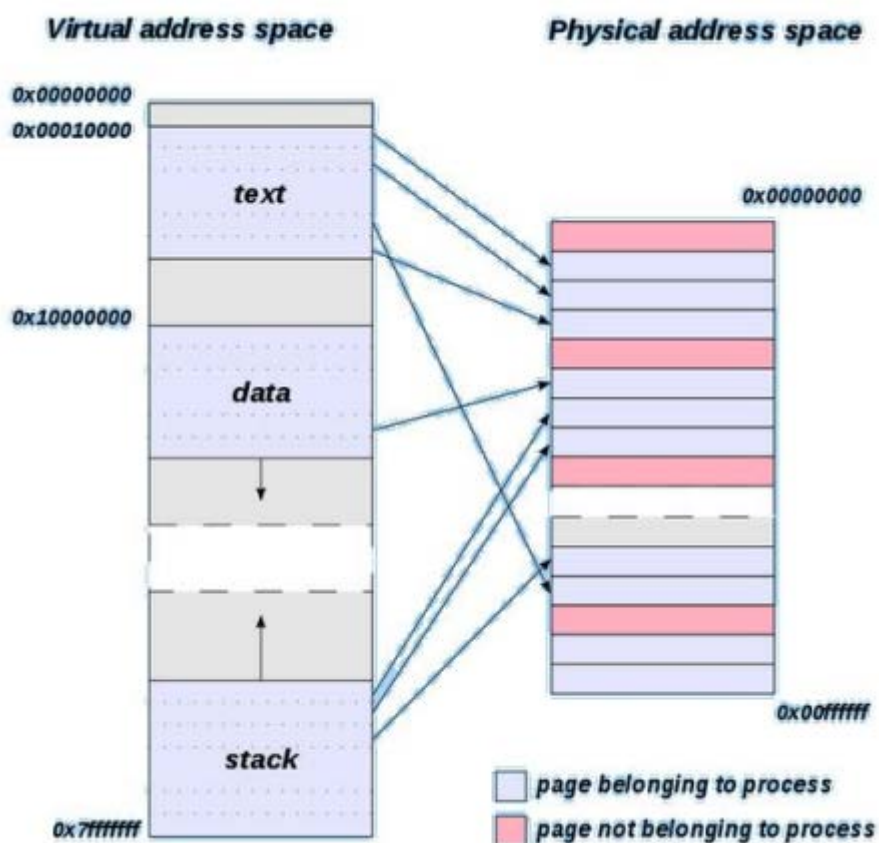


Рисунок 3.1 – Візуалізація пам'яті в C++

Переваги мови програмування C++ вражають своєю різноманітністю та потужністю. По-перше, C++ славиться крос-платформовістю, що дозволяє запускати програми на різних операційних системах та інтерфейсах, надаючи розробникам гнучкість у виборі платформи для своїх проектів.

Крім того, важливою перевагою є ефективне керування пам'яттю в C++, оскільки вона не використовує автоматичних збирачів сміття. Це сприяє

оптимальній роботі програм та дозволяє розробникам більш точно контролювати використання ресурсів.

Сумісність з мовою C робить C++ відмінним вибором для написання інструкцій компілятору та виправлення низькорівневих помилок, завдяки чому розробка стає більш гнучкою та швидшою.

Ще однією вагомою перевагою є підтримка різних парадигм програмування: процедурного, функціонального та об'єктно-орієнтованого програмування, що дозволяє розробникам вибирати підхід, який найкраще відповідає конкретним потребам їх проекту.

Незважаючи на численні переваги, варто відзначити, що C++ має свої недоліки. Специфікація мови може виглядати складною для самостійного вивчення, а також можуть виникати проблеми з безпекою через об'єктно-орієнтований підхід. Слід також бути уважними щодо витоків пам'яті при використанні покажчиків, хоча існують механізми використання розумних вказівників, що спрощує роботу з пам'яттю.

Для візуалізації програм на C++ використовується GUI, а саме Qt – крос-платформове програмне забезпечення, що дозволяє створювати інтерфейси для настільних, мобільних та вбудованих систем. Початок розробки цього інструменту налічується ще з 90-х років.

Qt – це не лише бібліотека, але й повноцінний фреймворк, який не представляє собою мову програмування, але додає значну функціональність до мови C++. Він був розроблений для розширення мови C++ шляхом введення нових можливостей, зокрема, сигналів та слотів.

Перед етапом компіляції, метаоб'єктний компілятор (МОС) проаналізує вихідні файли, написані на Qt-extended C++, та генерує сумісний вихідний код на C++. Це дозволяє фреймворку та програмам, що його використовують, працювати з будь-яким стандартним компілятором C++, таким як Clang, GCC, MSVC та інші.

Однією з важливих переваг Qt є наявність окремого сайту від розробника з детальною документацією щодо структури роботи з

фреймворком. Крім того, на стартовому етапі розробки, вам доступні готові проекти, які значно полегшують початок створення власних програм чи бібліотек (рисунок 3.2).

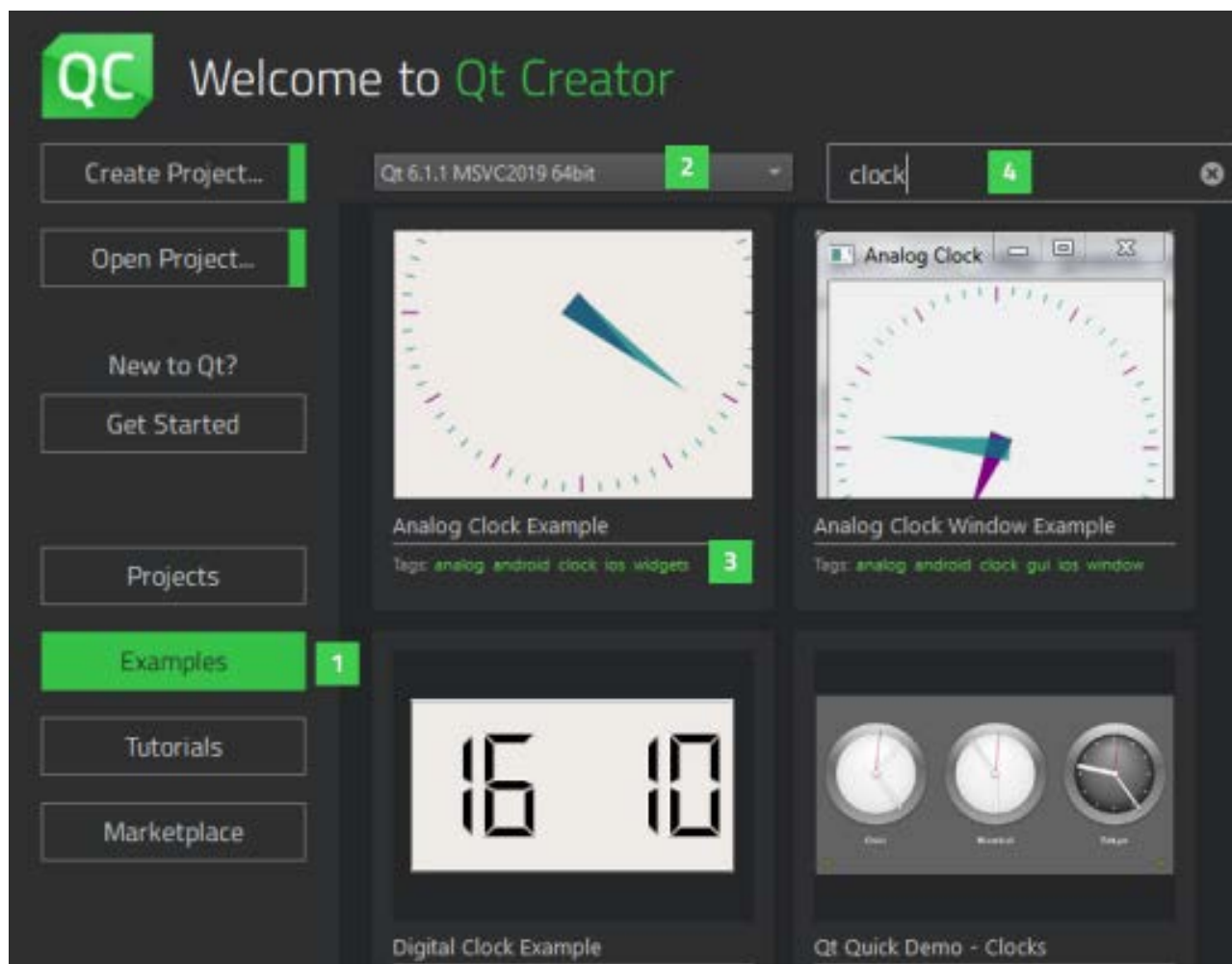


Рисунок 3.2 – середовище розробки Qt Creator

3.2 Розробка програмної частини

Основа програми – реалізований клас алгоритму Хаффмана (рис. 3.3)

Для ефективною модернізації нашого проекту важливо працювати з бітами, і для цього ми використовуємо стандартну бібліотеку C++ - `std::bitset<N>`. Цей шаблонний клас реалізує послідовність фіксованого розміру N-біт, що дозволяє нам зручно оперувати бітами. З особливим задоволенням повідомляємо, що в оновленні C++20 додані додаткові функції для роботи з бітами, такі як підрахунок кількості нулів та одиниць, вивертання бітів та інші зручні операції.

```

class Huffman{
public:
    ~Huffman();
    Node* generate_huffman_tree(const std::unordered_map<char,
                               std::uint64_t>& value);
    Node* generate_modified_huffman_tree(const std::unordered_map<char,
                                         std::uint64_t>& value);
    void print_huffman_tree(Node* root, std::stringstream& os,
                           Tree *prev = nullptr, bool is_left = false);
    uint64_t store_huffman_value(const Node *root, std::string &value);
    void print_huffman_codes(const Node *root, std::stringstream& ss);
    _NODISCARD std::uintmax_t create_file_huffman_code(const QString& file_name,
                                                       const char* file_name_new);

    Node* get_root();
    void destroy(Node* node);
private:
    std::string HuffmanValue[128];
    Node* huffman_tree;
    std::unordered_map<char, uint64_t> value_map;
};

```

Рисунок 3.3 – Реалізація алгоритму Хаффмана

Алгоритм побудови дерева Хаффмана, який використовується в нашому проекті, отримує на вхід клас `std::unordered_map`. У цьому класі ключами є символи, а значеннями - кількість входжень кожного символу у вхідному повідомленні. Цей метод, який був ретельно описаний у розділі 2 (рисунок 3.4), дозволяє нам ефективно створювати оптимальне дерево для подальшого використання в нашому проекті.

На зазначеній ілюстрації використано не просто чергу, а вектор, який в подальшому буде відсортований відносно частоти появи символів. Це фактично є пріоритетною чергою, але базується на іншій структурі даних. Цей підхід підкреслює важливість ефективного використання ресурсів для оптимальної реалізації алгоритму.

Нововведений у модернізованому алгоритмі Хаффмана метод призначений для ефективного стиснення даних. Основна логіка реалізована мовою програмування C++, яка володіє зручною функціональністю для роботи з бітами, що робить її ідеальним вибором для цього завдання.

```

Node *Huffman::generate_huffman_tree(const std::unordered_map<char, uint64_t> &value)
{
    value_map = std::move(value);
    auto store = utilities::sort_by_character_count(value_map);
    Node *one, *two, *parent;
    if (store.size() == 1)
        return Node::combine(store.back(), nullptr);

    while (store.size() > 2) {
        one = *(store.end() - 1);
        two = *(store.end() - 2);
        parent = Node::combine(one, two);
        store.pop_back();
        store.pop_back();
        store.push_back(parent);

        std::vector<Node*>::iterator it1 = store.end() - 2;
        while ((*it1)->count < parent->count && it1 != begin(store)) {
            --it1;
        }
        std::sort(it1, store.end(), utilities::comp);
    }
    one = *(store.end() - 1);
    two = *(store.end() - 2);

    huffman_tree = Node::combine(one, two);
    return huffman_tree;
}

```

Рисунок 3.4 – Алгоритм побудови дерева Хаффмана

Для генерації ключа необхідно обчислити розподіл частот кожного символу у вхідній послідовності, а потім відсортувати їх у лексикографічному порядку. Отримані частоти символів використовуються як значення ключа. Якщо довжина вхідного тексту має квадратний корінь, який є цілим числом, що дорівнює розміру ключа, формується квадратна матриця X . Однак слід зазначити, що добуток рядків і стовпців матриці X не завжди може дорівнювати кількості елементів у вхідних даних, позначеній як N . Процес генерації ключа складається з наступних кроків:

1. Спочатку вхідний текст сортується за зростанням. Потім визначається частота кожного символу, що зустрічається у вхідному тексті.
2. Значення ключа, що використовується, - це частота кожного символу, і вхідні дані перетворюються в матрицю X за певною процедурою. Розмір ключа визначає кількість стовпців у матриці X , тоді як кількість рядків визначається за допомогою рівняння (2.1).

3. Стовпці `matrix` проходять кроки зсуву на основі значення ключа.

```

_NODISCARD std::uintmax_t rle_compress(const char* file_name_open,
                                     const char* file_name_save)
{
    std::fstream file;
    file.open(file_name_open);
    std::ofstream outfile(file_name_save);

    char ch = 2;
    std::string str;
    int freq = 1;

    while (getline(file, str)) {
        for (std::size_t i = 0; i <= str.length(); ++i) {
            if (i==str.length()) {
                outfile << std::bitset<8>(ch).to_string() <<
                    std::bitset<8>(freq).to_string();
                ch = file.peek();
                freq=0;
                break;
            }
            else if (ch == str[i])
                freq++;
            else
            {
                outfile << std::bitset<8>(ch).to_string() <<
                    std::bitset<8>(freq).to_string();
                ch = str[i];
                freq = 1;
            }
        }
    }

    outfile.close();
    file.close();
    return std::filesystem::file_size(file_name_save);
}

```

Рисунок 3.5 – Алгоритм RLE

У рамках розробки було також створено декілька класів, спрямованих на полегшення роботи з файлами, відображення дерева та таблиці відповідно до оригінального бітового коду за ASCII, а також згенерованого коду за допомогою наших алгоритмів. Ці класи забезпечують зручний та інтуїтивно зрозумілий інтерфейс для взаємодії з різними елементами нашого проекту.

```

_NODISCARD std::uintmax_t write_binary_code(const QString& file_name,
                                           const char* file_name_new){
    std::fstream file;
    file.open(file_name.toLocal8Bit().data());

    std::ofstream outfile (file_name_new);
    unsigned char ch = 2;
    while (file >> std::noskipws >> ch){
        outfile << std::bitset<8>(ch).to_string();
    }
    outfile.close();
    file.close();
    return std::filesystem::file_size(file_name_new);
}

```

Рисунок 3.6 – Бітове представлення чисел

```

_NODISCARD std::uintmax_t lzw_compress(const char* file_name_open,
                                       const char* file_name_save) {
    std::vector<int> result;
    std::fstream file;
    file.open(file_name_open);
    std::ofstream outfile(file_name_save);

    int dict_size = 256;
    std::map<std::string,int> dictionary;
    for (int i = 0; i < 256; i++)
        dictionary[std::string(1, i)] = i;

    std::string w;
    char ch;
    while (file >> std::noskipws >> ch){
        std::string wc = w + ch;
        if (dictionary.count(wc))
            w = wc;
        else {
            result.emplace_back(dictionary[w]);
            dictionary[wc] = dict_size++;
            w = std::string(1, ch);
        }
    }

    if (!w.empty())
        result.emplace_back(dictionary[w]);

    for (auto& a: result)
        outfile << std::bitset<8>(a).to_string();

    outfile.close();
    file.close();
    return std::filesystem::file_size(file_name_save);
}

```

Рисунок 3.7 – Алгоритм LZW

Це важливий етап в розробці, оскільки дозволяє не лише взаємодіяти з вихідними та згенерованими даними, але й ефективно відображати їх структуру та характеристики. Візуалізація дерева та таблиці надає користувачеві зрозумілу графічну репрезентацію роботи алгоритмів стиснення, що полегшує аналіз та відлагодження.

3.3. Опис інтерфейсу програми

Завдяки використанню фреймворка Qt у нашій програмі, автоматично створюється файл з розширенням .ui, що містить інформацію про графічний інтерфейс користувача. Для запуску програми необхідно відкрити відповідний файл з розширенням .exe, що автоматично відкриє головне меню. Це головне меню є одним з .ui файлів, який забезпечує зручний та інтуїтивно зрозумілий інтерфейс для взаємодії з користувачем (рис 3.8). Цей підхід дозволяє легко навігувати та використовувати різні функціональності нашого застосунку.

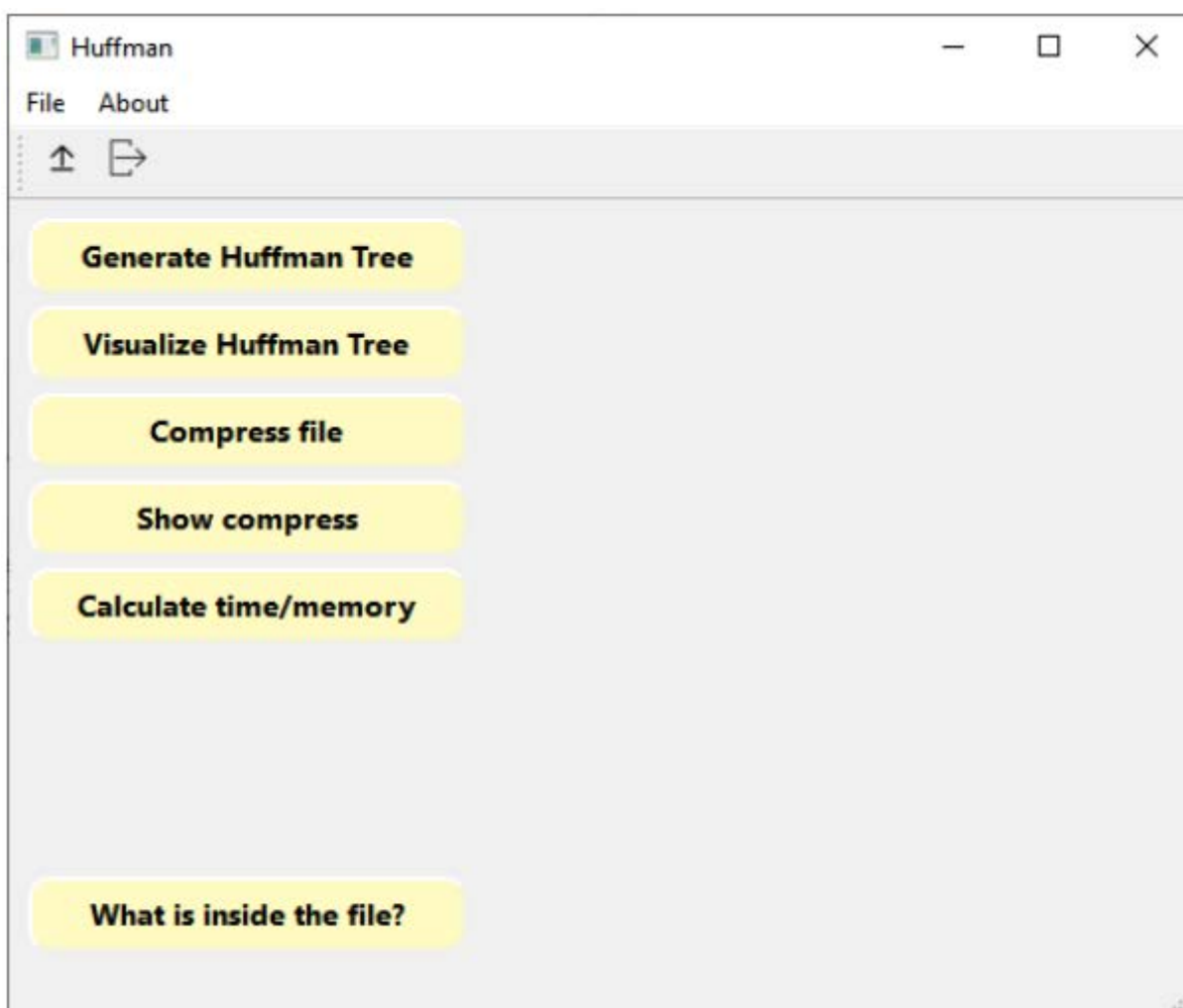


Рисунок 3.8 – Головне вікно роботи програми

Для ініціації роботи в даному застосунку, спершу необхідно завантажити текстовий файл, після чого ви можете взяти на себе контроль над алгоритмами, натискавши на кнопки «Generate Huffman Tree» та «Compress file». Це ініціює використання алгоритму стиснення, що зменшить розмір файлу.

Для отримання деталей про частоту кожного символу у файлі та відповідні двійкові коди, ви можете скористатися опцією «Show compress» (рис3.9). Ця можливість надає користувачеві важливу інформацію про результати роботи алгоритмів та структуру стисненого файлу.

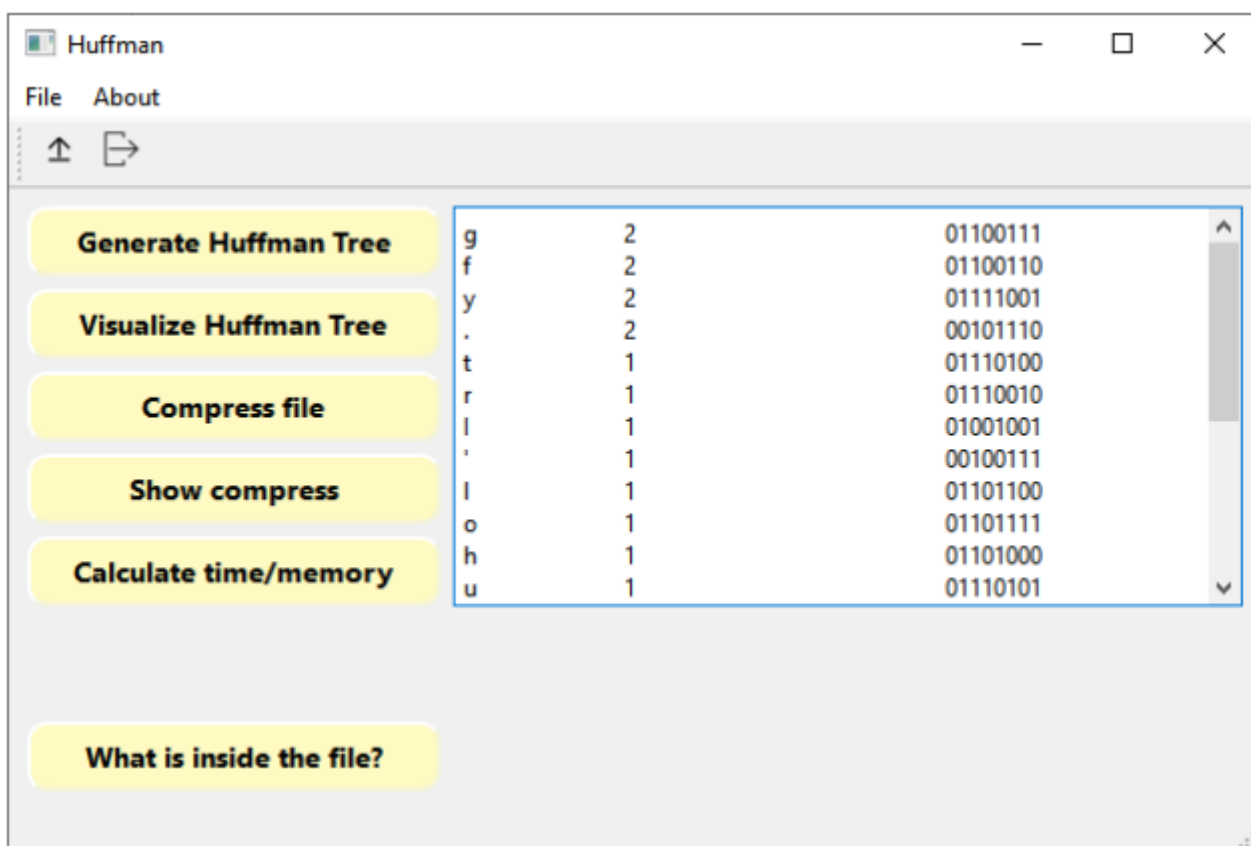


Рисунок 3.9 – Частота появи кожного символу та двійковий код за ASCII

Щоб візуалізувати дерево, необхідно скористатися опцією «Visualize Huffman Tree», яку можна активувати натисканням відповідної кнопки. Для здійснення цієї функціональності була розроблена спеціальна функція під назвою `print_huffman_tree`. Ця функція дозволяє пройтися по структурі дерева та вивести відображення його складу, включаючи символи та їх частоту (рис 3.10).

Використання такої візуалізації надає користувачеві зрозумілу та компактну інформацію про структуру Huffman-дерева, що може бути важливою для аналізу та налагодження алгоритмів стиснення.

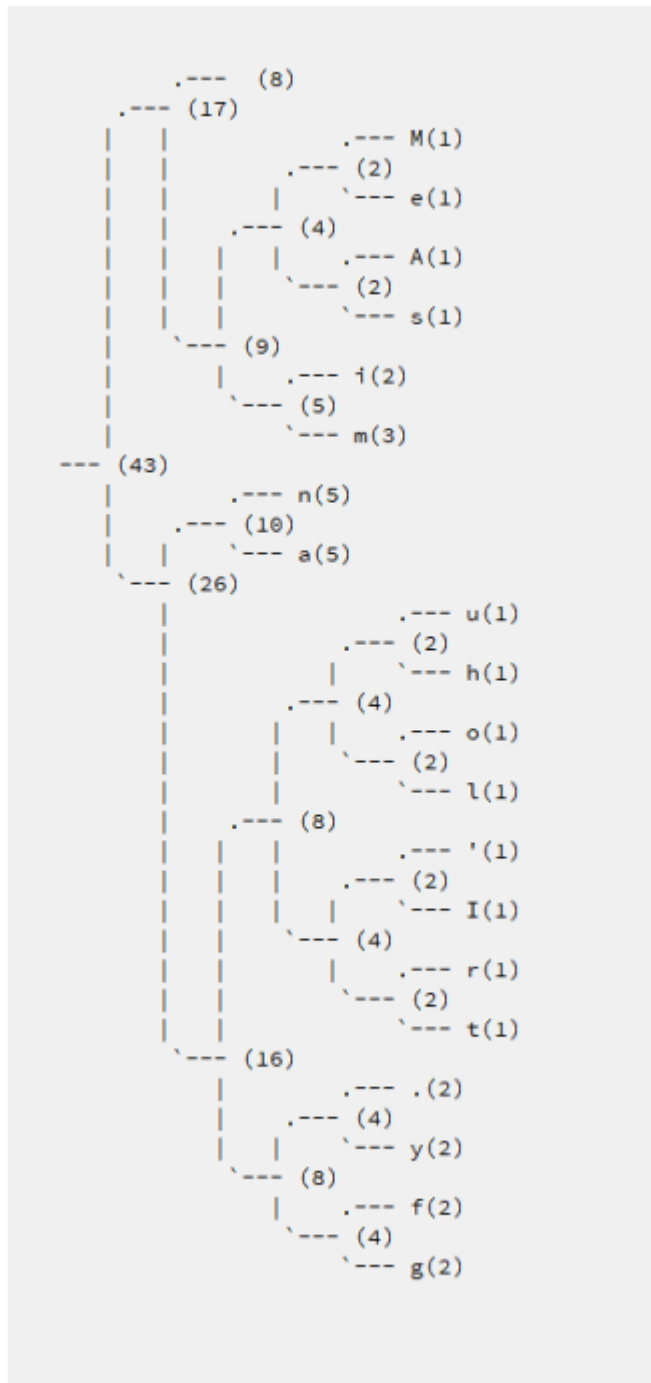


Рисунок 3.10 – Візуалізація дерева Хаффмана

Для отримання інформації про час та використану пам'ять, скористайтеся опцією «Calculate time/memory». Пам'ять, яка враховується, не обмежується лише тією, яка використовується під час роботи програми, але включає також інформацію про розмір файлу перед та після стиснення.

Для точного вимірювання часу використовується об'єкт `QElapsedTimer`, який розпочинає відлік перед побудовою дерева і завершує його після застосування кодування, тобто після завершення процесу стиснення (рис 3.11). Це дозволяє здійснювати точні та надійні вимірювання для подальшого аналізу продуктивності алгоритму.

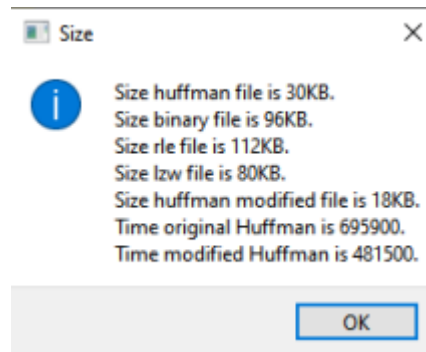


Рисунок 3.11 - наведено вивід результатів роботи алгоритмів.

Де кожному конкретному файлу призначено розмір всіх вихідних файлів, згенерованих під час виконання програми. В цьому виводі відображені різні види алгоритмів стиснення для подальшого порівняння їх ефективності. Такий підхід надає користувачеві повний обзор ефективності різних методів стиснення та впливу цих алгоритмів на розмір вихідних файлів.

Зазначено, що при спробі побудови дерева для порожнього файлу, вміст якого можна перевірити за допомогою кнопки «What is inside the file?» (рисунок 3.12). Це дозволяє вам зручно провести тестування алгоритмів у випадку особливого вхідного вмісту, а також надає корисний механізм для перевірки та аналізу структури порожніх файлів.

У випадку, коли вміст файлу є нульовим або файл не було завантажено, алгоритм не буде активований. Це важливо враховувати, оскільки забезпечує відсутність непотрібної роботи алгоритму в таких ситуаціях.

З метою полегшення роботи з програмою, де важливо працювати з файлами, був розроблений функціонал, який дозволяє відображати список завантажених файлів. Це створює можливість швидкого доступу до раніше використаних файлів та повторного їх завантаження за допомогою відповідної функції (рисунок 3.13).

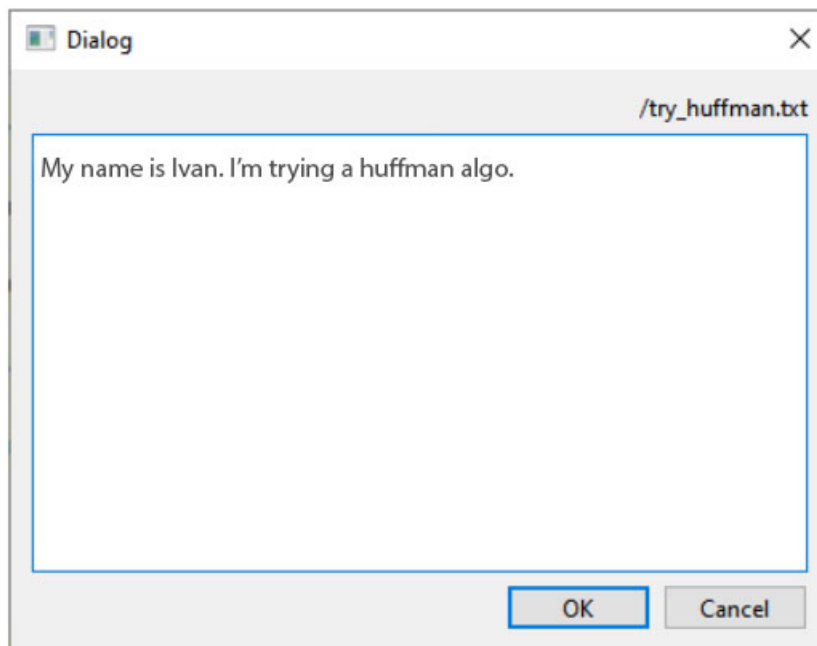


Рисунок 3.12 – Вивід файлу

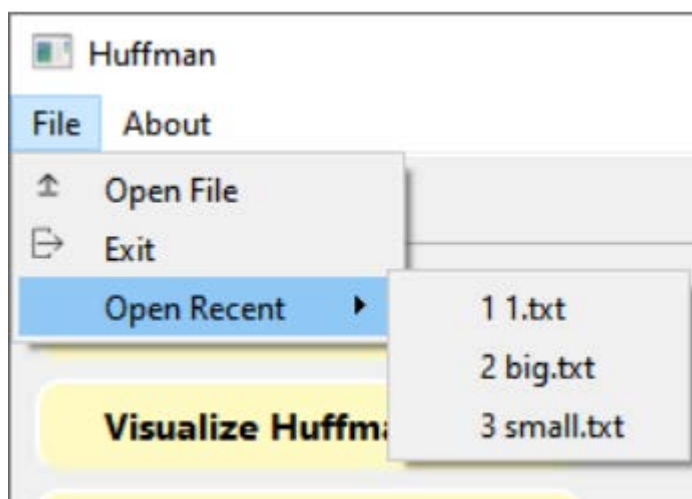


Рисунок 3.13 – Recent files

3.4 Тестування програми

Тестування даної програми можна розділити на перевірку роботи програми та аналіз результатів за кількісними та емпіричними показниками, такими як час та використання пам'яті. Це включає порівняння продуктивності алгоритму стиснення з іншими поширеними алгоритмами, такими як RLE, LZW та бітове представлення ASCII.

З метою отримання максимальної картини цих показників рекомендується використовувати різні вхідні файли, оскільки наявність різних

даних може значно вплинути на результати, враховуючи, що у нас єдиний комп'ютер та компілятор.

Результати порівняння ефективності алгоритму стиснення даних за допомогою використання бітів, які не задіяні зображено в таблиці 3.1.

Таблиця 3.1 - Порівняльна таблиця вихідного розміру файлів

Тестовий файл	Кількість символів	Розмір файлу КВ, алгоритм Хаффмана	Розмір файлу КВ, модифікований алгоритм Хаффмана	Розмір файлу КВ, бітове представлення	Розмір файлу КВ, алгоритм RLE	Розмір файлу КВ, алгоритм LZW
Файл 1	12	60	48	96	112	80
Файл 2	43	321	257	344	672	328
Файл 3	7582	35067	34719	60656	118720	24296
Файл 4	17753	80883	80468	142024	275344	48792

Порівнюючи розміри вихідних файлів, можна визначити, що модифікований алгоритм стиснення займає менше місця, ніж його оригінальний еквівалент. Проте, важливо відзначити, що при обробці невеликих файлів із невеликою кількістю символів, наприклад, 12, відсоткова різниця становить приблизно 20 відсотків. З іншого боку, у випадку обробки великих файлів, коли різниця у відсотках зменшується до близько 0.5%, це пояснюється тим, що модифікації стосуються саме обходу дерева, впливаючи на розмір структури при кодуванні та декодуванні. Тому для повного розуміння впливу модифікацій слід порівняти розміри самих дерев, що відображено в таблиці 3.2.

Аналізуючи отримані результати, можна зазначити, що чим більший розмір файлу, тим більший відсоток стиснення досягнуто. Особливо це помітно у випадку четвертого файлу, де нам вдалося стиснути майже в половину об'єму. Під час опису алгоритму та його модифікації ми передбачали, що різниця в розмірі буде в $(7n + 2)$ біта, що є майже вдвічі

менше, ніж оригінальний алгоритм (Файл 1 – 40%, Файл 2 – 43%, Файл 3 – 52%, Файл 4 – 49%). Результати експерименту підтверджують наші прогнози.

Таблиця 3.2 – Розмір дерева Хаффмана

Тестовий файл	Кількість символів	Розмір дерева Хаффмана, КВ	Розмір модифікованого дерева Хаффмана, КВ
Файл 1	12	30	18
Файл 2	43	147	83
Файл 3	7582	666	318
Файл 4	17753	850	435

Однак, для повноцінної оцінки продуктивності, важливо розглянути також таблицю, яка містить дані про час виконання алгоритмів Хаффмана (таблиця 3.3).

Таблиця 3.3 - Порівняльна таблиця часу роботи алгоритмів

Тестовий файл	Кількість символів	Час роботи алгоритму, 10^{-6} с	Час роботи модифікованого алгоритму, 10^{-6} с
Файл 1	12	796,6	526,9
Файл 2	43	844	589,3
Файл 3	2	3056,1	2778,3
Файл 4	2	6146,3	5585,9

Під час аналізу отриманих результатів можна визначити, що час виконання дійсно зменшився, як це було передбачено. Зменшення розміру дерева призвело до зменшення часу, витраченого на його обхід, що відображено на графіку відношення кількості бітів до кожного файлу для кожного алгоритму (графік 1).

Графік відображає ефективність стиснення для кожного файлу порівняно з різними алгоритмами. Видно, як модифікований алгоритм

проявляє себе в зменшенні кількості бітів для кожного файлу, підтверджуючи його високу ефективність.

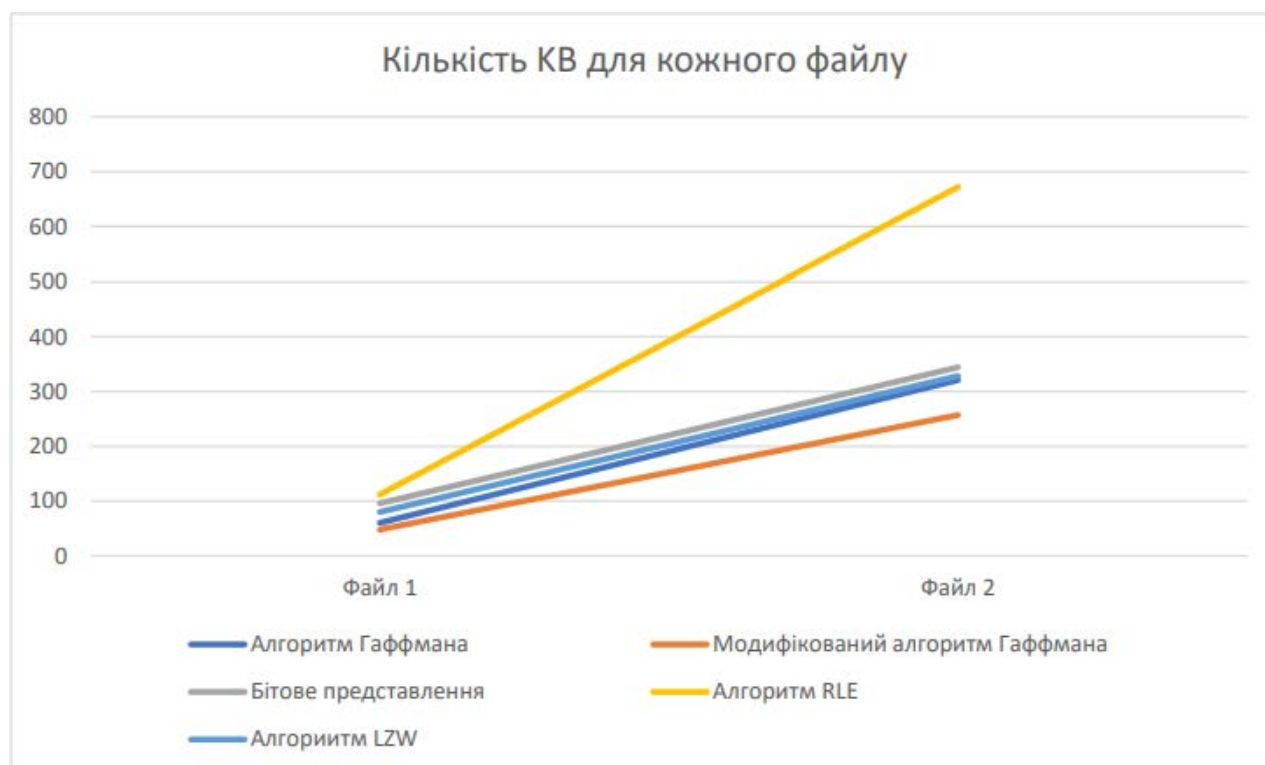


Рисунок 4.1 – Кількість KB, що займає кожен файл

При ретельному аналізі графіка стає очевидним, що алгоритм Хаффмана, так само як і алгоритм LZW, ефективно стискають дані, проте застосування нашої модифікації спричинило ще більше зменшення розміру файлів у кількості кілобайтів (KB).

У порівнянні з алгоритмом RLE спостерігається велика різниця. Це може пояснюватися тим, що в текстах англійської мови рідко виникає повторення літер, що створює надлишковість при стисненні. Для оптимізації RLE можна використовувати метод Берроуза-Вілера, який змінює порядок букв у реченнях і дозволяє ефективніше стискувати файли.

Під час визначення ентропії використовувалася функція entropy (рис. 3.1), що дозволяє кількісно оцінити ступінь невизначеності та передбачуваності даних у файлі.

```
double Huffman::entropy()
{
    double entropy = 0, probability = 0;
    for (auto i = 0; i < 127; ++i) {
        if (HuffmanValue[i].size()) {
            unsigned char ch = i;
            if (value_map[ch] == 0)
                continue;
            probability = (double)value_map[ch] / value_map.size();
            entropy += probability * log2(1.0 / probability);
        }
    }
    return entropy;
}
```

Рисунок 3.1 – Реалізація розрахунка ентропії

Для Файлу 1 теоретична ентропія – 2,46, практична – 2,9183.

4 ЕКОНОМІЧНА ЧАСТИНА

Виконання науково-дослідної роботи завжди передбачає отримання певних результатів і вимагає відповідних витрат. Результати виконаної роботи завжди дають нам нові знання, які в подальшому можуть бути використані для удосконалення та/або розробки (побудови) нових, більш продуктивних зразків техніки, процесів та програмного забезпечення.

Дослідження на тему «Телемедична система передачі та попередньої обробки медичних даних» може бути віднесено до фундаментальних і пошукових наукових досліджень і спрямоване на вирішення наукових проблем, пов'язаних з практичним застосуванням. Основою таких досліджень є науковий ефект, який виражається в отриманні наукових результатів, які збільшують обсяг знань про природу, техніку та суспільство, які розвивають теоретичну базу в тому чи іншому науковому напрямку, що дозволяє виявити нові закономірності, які можуть використовуватися на практиці.

Для цього випадку виконаємо такі етапи робіт:

- 1) здійснимо проведення наукового аудиту досліджень, тобто встановлення їх наукового рівня та значимості;
- 2) проведемо планування витрат на проведення наукових досліджень;
- 3) здійснимо розрахунок рівня важливості наукового дослідження та перспективності, визначимо ефективність наукових досліджень.

4.1 Оцінювання наукового ефекту

Основними ознаками наукового ефекту науково-дослідної роботи є новизна роботи, рівень її теоретичного опрацювання, перспективність, рівень розповсюдження результатів, можливість реалізації. Науковий ефект НДР на тему «Телемедична система передачі та попередньої обробки медичних даних» можна охарактеризувати двома показниками: ступенем наукової новизни та рівнем теоретичного опрацювання.

Значення показників ступеня новизни і рівня теоретичного опрацювання науково-дослідної роботи в балах наведені в табл. 4.1 та 4.2.

Таблиця 4.1 – Показники ступеня новизни науково-дослідної роботи виставлені експертами

Ступінь новизни	Характеристика ступеня новизни	Значення ступеня новизни, бали		
		Експерти (ПІБ, посада)		
		1	2	3
Принципово нова	Робота якісно нова за постановкою задачі і ґрунтується на застосуванні оригінальних методів дослідження. Результати дослідження відкривають новий напрям в даній галузі науки і техніки. Отримані принципово нові факти, закономірності; розроблена нова теорія. Створено принципово новий пристрій, спосіб, метод	0	0	0
Нова	Отримана нова інформація, яка суттєво зменшує невизначеність наявних значень (по-новому або вперше пояснені відомі факти, закономірності, впроваджені нові поняття, розкрита структура змісту). Проведено суттєве вдосконалення, доповнення і уточнення раніше досягнутих результатів	0	42	0
Відносно нова	Робота має елементи новизни в постановці задачі і методах дослідження. Результати дослідження систематизують і узагальнюють наявну інформацію, визначають шляхи подальших досліджень; вперше знайдено зв'язок (або знайдено новий зв'язок) між явищами. В принципі відомі положення розповсюджені на велику кількість об'єктів, в результаті чого знайдено ефективне рішення. Розроблені більш прості способи для досягнення відомих результатів. Проведена часткова раціональна модифікація (з ознаками новизни)	39	0	40
Традиційна	Робота виконана за традиційною методикою. Результати дослідження мають інформаційний характер. Підтверджені або поставлені під сумнів відомі факти та твердження, які потребують перевірки. Знайдено новий варіант рішення, який не дає суттєвих переваг в порівнянні з існуючим	0	0	0
Не нова	Отримано результат, який раніше зафіксований в інформаційному полі, та не був відомий авторам	0	0	0
Середнє значення балів експертів		40,3		

Згідно отриманого середнього значення балів експертів ступінь новизни характеризується як нова, тобто отримана нова інформація, яка суттєво зменшує невизначеність наявних знань (по-новому або вперше пояснені відомі факти, закономірності, впроваджені нові поняття, розкрита структура змісту) та проведено суттєве вдосконалення, доповнення і уточнення раніше досягнутих результатів.

Таблиця 4.2 – Показники рівня теоретичного опрацювання науково-дослідної роботи виставлені експертами

Характеристика рівня теоретичного опрацювання	Значення показника рівня теоретичного опрацювання, бали		
	Експерт (ПІБ, посада)		
	1	2	3
Відкриття закону, розробка теорії	0	0	0
Глибоке опрацювання проблеми: багатоаспектний аналіз зв'язків, взаємозалежності між фактами з наявністю пояснень, наукової систематизації з побудовою евристичної моделі або комплексного прогнозу	64	67	65
Розробка способу (алгоритму, програми), пристрою, отримання нової речовини	0	0	0
Елементарний аналіз зв'язків між фактами та наявною гіпотезою, класифікація, практичні рекомендації для окремого випадку тощо	0	0	0
Опис окремих елементарних фактів, викладення досвіду, результатів спостережень, вимірювань тощо	0	0	0
Середнє значення балів експертів	65,3		

Згідно отриманого середнього значення балів експертів рівень теоретичного опрацювання науково-дослідної роботи характеризується як глибоке опрацювання проблеми: багатоаспектний аналіз зв'язків, взаємозалежності між фактами з наявністю пояснень, наукової систематизації з побудовою евристичної моделі або комплексного прогнозу.

Показник, який характеризує рівень наукового ефекту, визначаємо за формулою [22]

$$E_{\text{нау}} = 0,6 \cdot k_{\text{нов}} + 0,4 \cdot k_{\text{теор}}, \quad (4.1)$$

де $k_{\text{нов}}, k_{\text{теор}}$ - показники ступеня новизни та рівня теоретичного опрацювання науково-дослідної роботи, $k_{\text{нов}} = 40,3, k_{\text{теор}} = 65,3$ балів;

$0,6$ та $0,4$ – питома вага (значимість) показників ступеня новизни та рівня теоретичного опрацювання науково-дослідної роботи.

$$E_{\text{нау}} = 0,6 \cdot k_{\text{нов}} + 0,4 \cdot k_{\text{теор}} = 0,6 \cdot 40,3 + 0,4 \cdot 65,33 = 50,33 \text{ балів.}$$

Визначення характеристики показника $E_{\text{нау}}$ проводиться на основі висновків експертів виходячи з граничних значень, які наведені в табл. 4.3.

Таблиця 4.3 – Граничні значення показника наукового ефекту

Досягнутий рівень показника	Кількість балів
Високий	70...100
Середній	50...69
Достатній	15...49
Низький (помилкові дослідження)	1...14

Відповідно до визначеного рівня наукового ефекту проведеної науково-дослідної роботи на тему «Телемедична система передачі та попередньої обробки медичних даних», даний рівень становить 50,33 балів і відповідає статусу - середній рівень. Тобто у даному випадку можна вести мову про потенційну фактичну ефективність науково-дослідної роботи.

4.2 Розрахунок витрат на здійснення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Телемедична система передачі та попередньої обробки медичних даних», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.2.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою [12]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.2)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, $T_p=22$ дні.

$$Z_o = 16980,00 \cdot 32 / 22 = 24698,18 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.4 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	16980,00	771,82	32	24698,18
Інженер-програміст	16910,00	768,64	11	8455,00
Інженер-проектувальник телемедичної системи	16150,00	734,09	22	16150,00
Консультант (лікар-терапевт)	12300,00	559,09	5	2795,45
Всього				52098,64

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт НДР на тему «Телемедична система передачі та попередньої обробки медичних даних» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.3)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.4)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо $M_M=6700,00$ грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [12];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно $T_p = 22$ дн;

$t_{зм}$ – тривалість зміни, год.

$$C_i = 6700,00 \cdot 1,10 \cdot 1,35 / (22 \cdot 8) = 56,53 \text{ грн.}$$

$$Z_{p1} = 56,53 \cdot 12,00 = 678,38 \text{ грн.}$$

Таблиця 4.5 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Установка обладнання	12,00	2	1,10	56,53	678,38
Підготовка робочого місця розробника системи	8,00	3	1,35	69,38	555,03
Інсталяція програмного забезпечення	6,20	4	1,50	77,09	477,95
Формування бази даних	7,50	5	1,70	87,37	655,25
Налагодження компонентів телемедичної системи	6,00	5	1,70	87,37	524,20
Налагодження програмних блоків	11,00	4	1,50	77,09	847,97
Тестування програмного забезпечення	10,00	3	1,35	69,38	693,79
Всього					4432,56

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (4.5)$$

де $H_{\text{дод}}$ – норма нарахування додаткової заробітної плати. Прийmemo 10%.

$$Z_{\text{дод}} = (52098,64 + 4432,56) \cdot 10 / 100\% = 5653,12 \text{ грн.}$$

4.2.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{ood}) \cdot \frac{H_{zn}}{100\%} \quad (4.6)$$

де H_{zn} – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (52098,64 + 4432,56 + 5653,12) \cdot 22 / 100\% = 13680,55 \text{ грн.}$$

4.2.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Телемедична система передачі та попередньої обробки медичних даних».

Витрати на матеріали на даному етапі проведення досліджень в основному пов'язані з використанням моделей елементів та моделювання роботи і досліджень за допомогою комп'ютерної техніки та створення експериментальних математичних моделей або програмного забезпечення, тому дані витрати формуються на основі витратних матеріалів характерних для офісних робіт.

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{ej}, \quad (4.7)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

C_{ej} – вартість відходів j -го найменування, грн/кг.

$$M_1 = 3 \cdot 198,00 \cdot 1,02 - 0 \cdot 0 = 605,88 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.6 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір 500 80 г\м	198,00	3	0	0	605,88
Папір для записів 100 70 г\м	113,00	3	0	0	345,78
Органайзер офісний	167,00	3	0	0	511,02
Набір офісний (канцелярське приладдя)	199,00	3	0	0	608,94
Картридж для принтера	1315,00	1	0	0	1341,30
Диск оптичний CD-R	28,00	5	0	0	142,80
Flesh-пам'ять 64 GB	229,00	1	0	0	233,58
Тека для паперів	95,00	5	0	0	484,50
Всього					4273,80

4.2.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (K_6), які використовують при проведенні НДР на тему «Телемедична система передачі та попередньої обробки медичних даних», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$K_6 = \sum_{j=1}^n H_j \cdot C_j \cdot K_j \quad (4.8)$$

де H_j – кількість комплектуючих j -го виду, шт.;

C_j – покупна ціна комплектуючих j -го виду, грн;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$).

$$K_6 = 1 \cdot 3820,00 \cdot 1,02 = 3896,40 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Комутатор	1	3820,00	3896,40
Відеокамера HD	1	1050,00	1071,00
Інтерфейсні блоки	2	735,00	1499,40
Всього			6466,80

4.2.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.і}} \cdot K_i, \quad (4.9)$$

де C_i – ціна придбання одиниці спецустаткування даного виду, марки, грн; $C_{\text{пр.і}}$ – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.; K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1,10\dots 1,12$); k – кількість найменувань устаткування.

$$B_{\text{спец}} = 44800,00 \cdot 1 \cdot 1,03 = 46144,00 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.8 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Серверне обладнання ASUS540-UA-X	1	44800,00	46144,00
Цифровий отоскоп SyncVision i010TO Pго	1	32680,00	33660,40
Мобільний 5/10-ти провідний електрокардіограф "Bee-W-New	1	30000,00	30900,00
Всього			110704,40

4.2.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{\text{прог}} = \sum_{i=1}^k C_{\text{прог}} \cdot C_{\text{прог.і}} \cdot K_i, \quad (4.10)$$

де $C_{\text{прог}}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{\text{прог.і}}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1, 10 \dots 1, 12$);

k – кількість найменувань програмних засобів.

$$B_{\text{прог}} = 9650,00 \cdot 1 \cdot 1,02 = 9843,00 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.9 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Математичне середовище MatLab (Deep Learning Toolbox)	1	9650,00	9843,00
Прикладне ПЗ Mathematica (Wolfram Neural Network Framework)	1	7690,00	7843,80
Всього			17686,80

4.2.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації.

Проведені розрахунки зведемо до таблиці.

Таблиця 4.10 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер розробника ПЗ	43699,00	2	2	3641,58
Персональний комп'ютер інженера-дослідника	28380,00	2	2	2365,00
Робоче місце інженера-програміста	9210,00	5	2	307,00
Робоче місце інженера-дослідника	9250,00	5	2	308,33
Пристрої передачі даних	6560,00	4	2	273,33
Оргтехніка	7680,00	5	2	256,00
Приміщення лабораторії розробки ПЗ	432000,00	35	2	2057,14
ОС Windows	6500,00	2	2	541,67
Прикладний пакет Microsoft Office	6510,00	2	2	542,50
Всього				10292,56

За формулою:

$$A_{обл} = \frac{Ц_{об}}{T_e} \cdot \frac{t_{вик}}{12}, \quad (4.11)$$

де $Ц_{об}$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

T_e – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (43699,00 \cdot 2) / (2 \cdot 12) = 3641,58 \text{ грн.}$$

4.2.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e)

Проведені розрахунки зведемо до таблиці.

Таблиця 4.11 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер розробника ПЗ	0,42	240,0	756,00
Персональний комп'ютер інженера-дослідника	0,05	160,0	60,00
Робоче місце інженера-програміста	0,12	100,0	90,00
Робоче місце інженера-дослідника	0,10	160,0	120,00
Пристрої передачі даних	0,05	80,0	30,00
Оргтехніка	0,45	3,2	10,80
Серверне обладнання ASUS540-UA-X	0,32	100,0	240,00
Цифровий отоскоп SyncVision i01OTO Pro	0,06	80,0	36,00
Всього			1342,80

Розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{eni}}{\eta_i}, \quad (4.12)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 7,50$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,42 \cdot 240,0 \cdot 7,50 \cdot 0,95 / 0,97 = 756,00 \text{ грн.}$$

4.2.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Телемедична система передачі та попередньої обробки медичних даних» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cb} = (Z_o + Z_p) \cdot \frac{H_{cb}}{100\%}, \quad (4.13)$$

де H_{cb} – норма нарахування за статтею «Службові відрядження», прийmemo $H_{cb} = 20\%$.

$$B_{cb} = (52098,64 + 4432,56) \cdot 20 / 100\% = 11306,24 \text{ грн.}$$

4.2.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (4.14)$$

де H_{cn} – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo $H_{cn} = 30\%$.

$$B_{cn} = (52098,64 + 4432,56) \cdot 30 / 100\% = 16959,36 \text{ грн.}$$

4.2.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_g = (Z_o + Z_p) \cdot \frac{H_{ig}}{100\%}, \quad (4.15)$$

де H_{ig} – норма нарахування за статтею «Інші витрати», прийmemo $H_{ig} = 50\%$.

$$I_g = (52098,64 + 4432,56) \cdot 50 / 100\% = 28265,60 \text{ грн.}$$

4.2.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.16)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo $H_{нзв} = 100\%$.

$$B_{нзв} = (52098,64 + 4432,56) \cdot 100 / 100\% = 56531,20 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Телемедична система передачі та попередньої обробки медичних даних» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = Z_o + Z_p + Z_{дод} + Z_n + M + K_{в} + B_{снец} + B_{прг} + A_{обл} + B_e + B_{св} + B_{сп} + I_{в} + B_{нзв}. \quad (4.17)$$

$$B_{заг} = 52098,64 + 4432,56 + 5653,12 + 13680,55 + 4273,80 + 6466,80 + 110704,40 + 17686,80 + 10292,56 + 1342,80 + 11306,24 + 16959,36 + 28265,60 + 56531,20 = 339694,43 \text{ грн.}$$

Загальні витрати ZB на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ZB = \frac{B_{заг}}{\eta}, \quad (4.18)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta=0,95$.

$$ZB = 339694,43 / 0,95 = 357573,09 \text{ грн.}$$

4.3 Оцінювання важливості та наукової значимості науково-дослідної роботи

Оцінювання та доведення ефективності виконання науково-дослідної роботи фундаментального чи пошукового характеру є достатньо складним процесом і часто базується на експертних оцінках, тому має вірогідний характер.

Для обґрунтування доцільності виконання науково-дослідної роботи на тему «Телемедична система передачі та попередньої обробки медичних даних» використовується спеціальний комплексний показник, що враховує

важливість, результативність роботи, можливість впровадження її результатів у виробництво, величину витрат на роботу.

Комплексний показник K_p рівня науково-дослідної роботи може бути розрахований за формулою:

$$K_p = \frac{I^n \cdot T_C \cdot R}{B \cdot t}, \quad (4.19)$$

де I – коефіцієнт важливості роботи. Прийmemo $I = 4$;

n – коефіцієнт використання результатів роботи; $n = 0$, коли результати роботи не будуть використовуватись; $n = 1$, коли результати роботи будуть використовуватись частково; $n = 2$, коли результати роботи будуть використовуватись в дослідно-конструкторських розробках; $n = 3$, коли результати можуть використовуватись навіть без проведення дослідно-конструкторських розробок. Прийmemo $n = 2$;

T_C – коефіцієнт складності роботи. Прийmemo $T_C = 3$;

R – коефіцієнт результативності роботи; якщо результати роботи плануються вище відомих, то $R = 4$; якщо результати роботи відповідають відомому рівню, то $R = 3$; якщо нижче відомих результатів, то $R = 1$. Прийmemo $R = 3$;

B – вартість науково-дослідної роботи, тис. грн. Прийmemo $B = 357573,09$ грн;

t – час проведення дослідження. Прийmemo $t = 0,17$ років, (2 міс.).

Визначення показників I , n , T_C , R , B , t здійснюється експертним шляхом або на основі нормативів [12].

$$K_p = \frac{I^n \cdot T_C \cdot R}{B \cdot t} = 4^2 \cdot 3 \cdot 3 / 357,6 \cdot 0,17 = 2,42.$$

Якщо $K_p > 1$, то науково-дослідну роботу на тему «Телемедична система передачі та попередньої обробки медичних даних» можна вважати ефективною з високим науковим, технічним і економічним рівнем.

4.4 Висновок до розділу 4

Витрати на проведення науково-дослідної роботи на тему «Телемедична система передачі та попередньої обробки медичних даних» складають 357573,09 грн. Відповідно до проведеного аналізу та розрахунків рівень наукового ефекту проведеної науково-дослідної роботи на тему «Телемедична система передачі та попередньої обробки медичних даних» є середній, а дослідження актуальними, рівень доцільності виконання науково-дослідної роботи $K_p > 1$, що свідчить про потенційну ефективність з високим науковим, технічним і економічним рівнем.

ВИСНОВКИ

Дані, що зберігаються на фізичних носіях і передаються каналами зв'язку, часто містять багато надлишкової інформації, яку можна зменшити за допомогою методів стиснення, щоб заощадити місце і скоротити час, необхідний для передачі даних.

Потреба в адекватних заходах безпеки, таких як контроль секретних ключів у певних методах, викликає занепокоєння щодо вразливості даних до потенційних атак. Шифрування відіграє життєво важливу роль у захисті інформації та збереженні її конфіденційності за допомогою секретного ключа, який робить дані нечитабельними і незмінними. Алгоритми стиснення зазвичай використовують надлишковість і закономірності даних. Однак навіть незначна зміна даних може суттєво вплинути на ефективність стиснення.

Проаналізувавши поняття телемедицини можна зробити висновок, що це комплекс дій, технологій та заходів, що застосовуються під час надання медичної допомоги з використанням засобів дистанційного зв'язку для обміну інформацією. Крім цього, телемедичний напрям включає діджитал-освіту лікарів та пацієнтів, медичні інформаційні послуги та самообслуговування через цифрові комунікаційні технології.

Проаналізувавши складові телемедицини можна виділити шість складових які є основними в телемедицині: телемедичне консультування, біотелеметрія, дистанційне навчання, телеприсутність, телескринінг, домашня(персональна) телемедицина.

Одним головних аспектів в телемедицині є швидкість і стабільність інтернет зеднання, технічний аналіз багатосервісної волоконно-оптичної мережі для телемедицини показав що волоконе зеднання є одним із найшвидших і настаблінішим зеднанням і має великий період експлоатації.

Проведений аналіз архівації даних дозволяє використовувати ефективні методи зменшення обсягу інформації з метою оптимізації зберігання та

передачі. Застосування відповідних методів архівації може суттєво вплинути на продуктивність та ресурсозбереження в системах обробки даних.

Розгляд принципу роботи Хаффмана дозволяє зрозуміти, як ефективно використовувати кодування для стискування інформації. Застосування алгоритму Хаффмана в процесі стискування даних виявляється як перспективний засіб для оптимізації пропускної здатності мережі та зменшення обсягу зібраної інформації.

Проведений аналіз оцінки стиснення даних дозволяє об'єктивно оцінити ефективність застосованих методів стиснювання. Це важливий етап при виборі оптимальних стратегій для забезпечення оптимального балансу між ступенем стиснення та витратами ресурсів при обробці та передачі даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. В. Е. Кривоносов, С. М. Злепко, Т. А. Чернишова, та Я. І. Ярославський, “Алгоритм діагностики стану болтового струмопровідного з’єднання комп’ютерного томографа”, на IV Міжнар. наук. конф. Вимірювання, контроль та діагностика в технічних системах (ВКДТС-2017), Вінниця, 2017, с. 248-249.
2. Про затвердження нормативних документів щодо застосування телемедицини у сфері охорони здоров'я. Офіційний вебпортал парламенту України. URL: <https://zakon.rada.gov.ua/laws/show/z1400-15#Text> (дата звернення: 10.05.2023).
3. Grigsby et al. Analysis of expansion of access to care through telemedicine, report 4, study summary and recommendations for further research. Denver, CO : Center for Health Policy Research, 1994. 432 с.
4. What is telehealth? - CCHP. CCHP. URL: <https://www.cchpca.org/what-is-telehealth/?category=mobile-health> (дата звернення: 10.11.2023).
5. H1 2021 digital health funding: Another blockbuster year...in six months | Rock Health. Rock Health | We're powering the future of healthcare. Rock Health is a seed and early-stage venture fund that supports startups building the next generation of technologies transforming healthcare. URL: <https://rockhealth.com/insights/h1-2021-digital-health-funding-another-blockbuster-year-in-six-months/> (дата звернення: 10.11.2023).
6. Telemedicine Services Global Market Report 2021: COVID-19 Growth and Change to 2030. Research and Markets - Market Research Reports - Welcome. URL: <https://www.researchandmarkets.com/reports/5321435/telemedicine-services-global-market-report-2021> (дата звернення: 10.11.2023).
7. Telehealth: A quarter-trillion-dollar post-COVID-19 reality? / O. Bestsenyу та ін. McKinsey & Company. URL: <https://www.mckinsey.com/industries/healthcare-systems-and-services/our->

insights/telehealth-a-quarter-trillion-dollar-post-covid-19-reality (дата звернення: 10.11.2023).

8. American Medical Association. 2021 telehealth survey report. URL: <https://www.ama-assn.org/system/files/telehealth-survey-report.pdf> (дата звернення: 10.11.2023).

9. Control plane optimisation for an sdn-based WBAN framework to support healthcare applications / Hasan, K та ін. Sensors. 2020. № 20. С. 1–19.

10. Telesurgery QoS improvement over SDN based on a Type-2 fuzzy system and enhanced cuckoo optimization algorithm / Parsaei M. R та ін. Int J Commun Syst. 2020.

11. What is Software-Defined Networking (SDN)? | VMware Glossary. VMware. URL: <https://www.vmware.com/topics/glossary/content/software-defined-networking.html> (дата звернення: 10.11.2023).

12. An introduction to SDN. GitHub Pages. URL: <https://qmonnet.github.io/whirl-offload/2016/07/08/introduction-to-sdn/> (дата звернення: 10.11.2023).

13. What are smart contracts on blockchain? | IBM. IBM - Deutschland | IBM. URL: <https://www.ibm.com/topics/smart-contracts> (дата звернення: 10.11.2023).

14. Raja Wasim Ahmad, Khaled Salah, Raja Jayaraman, Ibrar Yaqoob, Samer Ellahham, and Mohammed Omar. Blockchain and COVID-19 Pandemic: Applications and Challenges. IEEE TechRxiv. 2020. С. 1–19. URL: <https://doi.org/10.36227/techrxiv.12936572> (дата звернення: 10.11.2023).

15. Sirina Keesara, M.D., Andrea Jonas, M.D., and Kevin Schulman, M.D. Covid-19 and Health Care's Digital Revolution | NEJM. New England Journal of Medicine. URL: <https://www.nejm.org/doi/10.1056/NEJMp2005835> (дата звернення: 10.11.2023).

16. Vivek Chauhan, Sagar Galwankar, Bonnie Arquilla, Manish Garg, Salvatore Di Somma, Ayman El-Menyar, Vimal Krishnan, Joel Gerber, Reuben Holland, Stanislaw P. Stawicki. Novel coronavirus (COVID-19): Leveraging

telemedicine to optimize care while minimizing exposures and viral transmission. *Journal of Emergencies Trauma and Shock*. 2020. Т. 13, № 1. С. 20–24.

17. D. Cherenkov, N. G. Kosulina, Y. I. Yaroslavskyy, "Justification of the electromagnetic impulse method destruction of insect pests", *Proc. SPIE 10808, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments* 2018, 108083P (1 October 2018), <https://doi.org/10.1117/12.2501665>.

18. К. Компанидзе, М. Тевторадзе, М. Мануков, М. М. Салдадзе и Е. Камкаmidзе, *Компьютерные сети телемедицины*. Тбилиси, Грузия: Издательский дом "Технический университет", 2009.

19. А. И. Царегородцев, "Эффективность использования телемедицины в северных районах", *Журнал телемедицины и электронного здравоохранения*, №1, с. 18-23, 2015. [21] А. В. Владзимирский, "Первичная телемедицинская консультация «пациент-врач»: первая систематизация методологии", *Журнал телемедицины и электронного здравоохранения*, – №2, с. 109-121, 2017.

20. Ehealth. Електронна система охорони здоров'я. [Електронний ресурс]. Доступно: <https://ehealth.gov.ua/>. Дата звернення: Жов. 29, 2018 168

21. А. В. Владимирский, В. Г. Климовицкий, Д. К. Калиновский, Р. В. Павлович, М. Ю. Сметанников, и Р. Л. Крутько, *Оборудование для телемедицинской деятельности лечебно-профилактических учреждений: методические рекомендации*. Донецк, Украина: ООО «Цифровая типография», 2007

22. В. И. Сырямкин, А. Н. Байков, А. Ш. Буреев, Д. С. Жданов и А. В. Осипов, "Разработка автоматизированной системы телемедицинских консультаций", *Бюллетень сибирской медицины*, № 6, с.125-130, 2011.

23. Стандарт DICOM 3.0 [Электронный ресурс]. Доступно: http://www.course-as.ru/download/pdf/DICOM/DICOM_Std.pdf. Дата обращения: Окт. 29, 2018.

24. АЛТ Украина Лтд. Архив медицинских изображений PACS. [Электронный ресурс]. Доступно: <https://www.mcmed.ua/ru/pacs>. Дата обращения: Окт. 29, 2018.
25. Integrating the Healthcare Enterprise (IHE). [Online]. Available: <https://www.ihe.net/>. Accessed on: Nov 09, 2018.
26. HL7 Standarts. [Online]. Available: <http://www.hl7.org/>. Accessed on: Nov 09, 2018.
27. L.N. Sutton, "PACS and diagnostic imaging service delivery - a UK perspective", Eur J Radiol. № 78(2), p. 243-249, 2011/
28. 1С: Медицина. Поликлиника. "Взаимодействие с PACS и медицинским оборудованием по стандарту DICOM 3". [Электронный ресурс]. Доступно: <https://solutions.1c.ru/catalog/clinic/dicom>. Дата обращения: Окт. 29, 2018.
29. Mallinckrodt Institute of Radiology. [Online]. Available: <https://www.mir.wustl.edu/education>. Accessed on: Nov 09, 2018.
30. К. Компанидзе, М. Тевторадзе, М. Мануков, М. М. Салдадзе и Е. Камкаmidзе, Компьютерные сети телемедицины. Тбилиси, Грузия: Издательский дом "Технический университет", 2009.
31. Общественное объединение - региональный благотворительный фонда содействия развитию информационно-коммуникационных технологий "Право и Интернет", Стандарт «Телемедицина. Общие положения», №1-2512/2002, Дек. 29, 2002. [Электронный ресурс]. Доступно: <http://www.russianlaw.net/law/self-regulation/z32/>. Дата обращения: Окт. 29, 2018.
32. Український телекомунікаційний портал, "Телемедицинские сети связи". [Электронный ресурс]. Доступно: <https://portaltele.com.ua/articles/network-technology/2010-11-03-08-25-47.html>. <https://medical.net.ua/images/img1567.png>. Дата обращения: Окт. 29, 2018.

33. Стэл – Компьютерные системы, “Пример построения региональной телемедицинской сети”. [Электронный ресурс]. Доступно: <http://old.stel.ru/telemedicine/idea/>. Дата обращения: Окт. 29, 2018.

34. I.S. Shergill, and A. Mohammed, Teleradiology: 21st century communication in surgery. Br J Hosp Med., № 72(5), p. 271-274, 2011.

35. Архивация медицинских диагностических изображений. Методические рекомендации. Москва: НПЦ Медрадиологии, 2014 [38] Д. К. Коноплянский, “PLC - передача данных по электрическим сетям. Последняя миля”, Вестник связи, № 5, с. 5-7, 2004.

36. А. В. Никифоров, “Технология PLC — телекоммуникации по сетям электропитания”, Сети и системы связи, № 5, с. 15-23, 2002.

37. Vincenzo Giordano, Flavia Gangale, Gianluca Fulli (JRC-IE), Manuel Sanchez Jimenez (DG ENER) and other, Smart Grid projects in Europe: lessons learned and current developments, Luxembourg: Publications Office of the European Union. 2011.

ДОДАТОК А
ПРОТОКОЛ
ПЕРЕВІРКИ НА ПЛАГІАТ
МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: «Телемедична система передачі та попередньої обробки медичних даних»

Тип роботи: магістерська кваліфікаційна робота

Підрозділ: кафедра БМІОЕС

Науковий керівник: Кожем'яко А.В., к.т.н., доцент каф. БМІОЕС


Показники звіту подібності:

Unicheck	
Оригінальність	95 %
Схожість	5 %


Аналіз звіту подібності (відмітити потрібне):

1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на доопрацювання.
3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений з повним звітом подібності, який був згенерований системою щодо роботи

Автор  Бондар І.Д.
(підпис)

Опис прийнятого рішення Робота допускається до захисту


Особа, відповідальна за перевірку  Тужанський С.Є.

ДОДАТОК Б
Технічне завдання

Міністерство освіти та науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри БМІОЕС

 к.т.н., доц. Коваль Л.Г.


(прізвище та ініціали)

« 18 » 09 2023 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

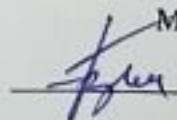
«Телемедична система передачі та попередньої обробки медичних даних»

 Науковий керівник

к.т.н., доц. каф. ОТ

Кожем'яко А.В.

виконав:

 магістрант 2 курсу,

Бондар І.Д.

Вінниця 2023

1. Підстава виконання магістерської кваліфікаційної роботи

1.1 Телемедична система передачі та обробки медичних даних — це перспективне напрямом, який включає в себе різноманітні технології для вдосконалення медичного обслуговування. Віддалений моніторинг пацієнтів дозволяє надавати здоров'я пацієнтів в реальному часі, забезпечуючи раннє виявлення змін у їхньому стані. Електронні медичні записи сприяють ефективній обмін інформацією між медичними установами та полегшують доступ лікарів до повної історії пацієнта. Телеконсультації та дистанційне консультування стають все більш доступними завдяки технологіям відеозв'язку, що робить медичні консультації більш зручними для пацієнтів, особливо в умовах віддалених регіонів або обмежень, таких як пандемія.

1.2 Наказ про затвердження теми МКР

2 Мета і призначенням МКР

2.1 Метою роботи є розширення функціональних можливостей телемедичної системи шляхом організації ефективного обміну медичними зображеннями у телемедичній мережі за допомогою методів стиснення даних. Особлива увага приділяється використанню методу Хаффмана для оптимізації розмірів зображень, що сприяє зменшенню обсягу передаваних даних та покращує продуктивність системи.

2.2 Призначення розробки — виконання магістерської кваліфікаційної роботи.

3 Вихідні дані для виконання МКР

Вихідні дані для виконання МКР: методи засновані на класичних алгоритмах проектування комп'ютерних мереж, пасивна оптична мережа засоби налаштування оптичних комутаторів та абонентських пристроїв.

4 Вимоги до виконання МКР

МКР повинна задовольняти такі вимоги:

- Забезпечити швидку передачу даних за допомогою стиснення методом Хаффмана.;
- провести моделювання та тестування системи;

5 Етапи МКР та очікувані результати

Етапи роботи та очікувані результати приведено в табл. А.1.

6 Матеріали, що подаються до захисту МКР

До захисту МКР подаються: пояснювальна записка МКР, ілюстративні та графічні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, відзив опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам.

Таблиця А.1 — Етапи МКР

№ з/п	Назва етапів МКР	Строк виконання етапів роботи	Примітка
1	Постановка задачі дослідження	18.09.2023	
2	Аналіз методів і систем керування розумними будинками	09.10.2023	
3	Розробка методу енергозабезпечення розумного будинка	09.11.2023	
4	Експериментальні дослідження системи	15.11.2023	
5	Економічний розділ	28.11.2023	
6	Попередній захист МКР	29.11.2023	
7	Нормоконтроль МКР	06.12.2023	
8	Рецензування МКР	07.12. 2023	
8	Остаточний Захист МКР	14.12.2023	

7 Порядок контролю виконання та захисту МКР

Виконання етапів розрахункової та графічної документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

8 Вимоги до оформлення МКР

8.1 При оформлюванні МКР використовуються:

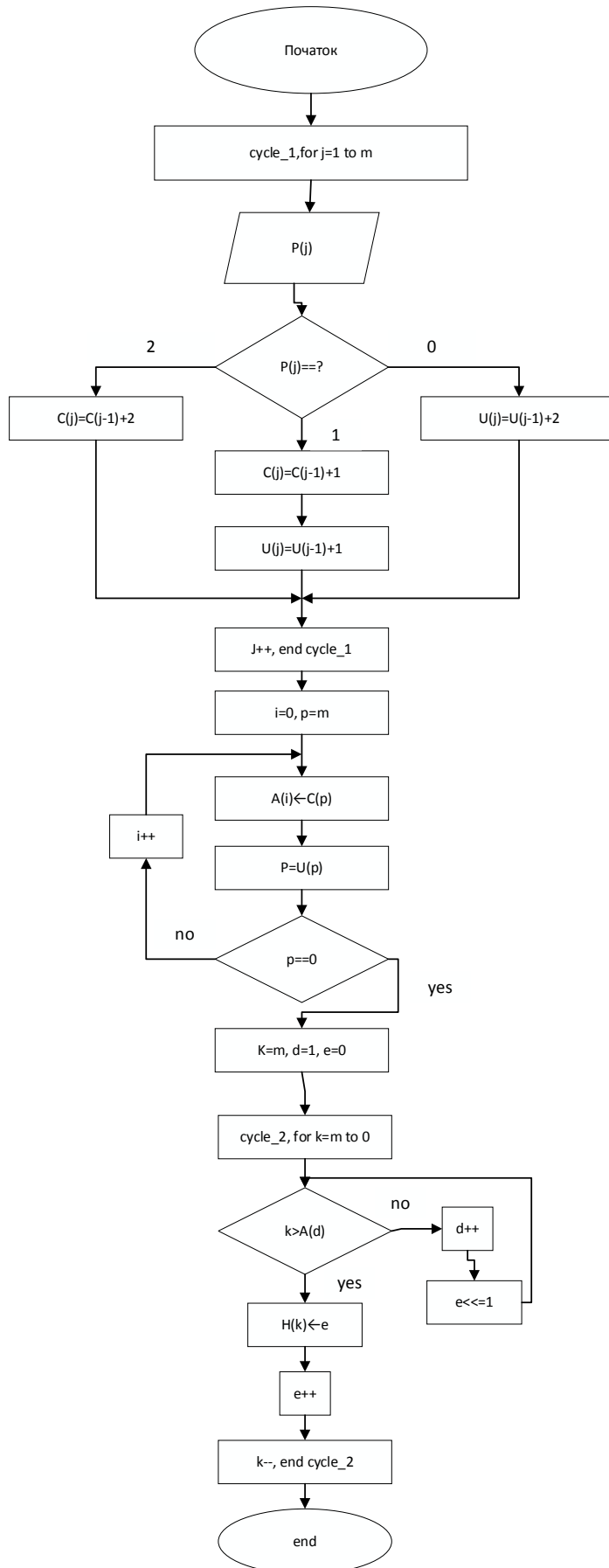
— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

8.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ–03.02.02 П.001.01:21.

ДОДАТОК В

Блок-схема



ДОДАТОК Г

Лістинг програми

```
#include <iostream>
#include <cstring>

using namespace std;

struct sym {
    unsigned char ch;
    float freq;
    char code[255];
    sym* left;
    sym* right;
};

void Statistics(const char* String, sym symbols[], int& k, int& kk);
sym* makeTree(sym* psym[], int k);
void makeCodes(sym* root);
void CodeHuffman(const char* String, char* BinaryCode, sym* root);
void DecodeHuffman(const char* BinaryCode, char* ReducedString, sym* root);

int main() {
    char String[1000];
    char BinaryCode[1000];
    char ReducedString[1000];

    cout << "Enter a string for encoding: ";
    cin >> String;

    sym symbols[256] = {0};
    int k = 0, kk = 0;

    Statistics(String, symbols, k, kk);

    sym* root = makeTree(symbols, k);
    makeCodes(root);

    CodeHuffman(String, BinaryCode, root);

    cout << "Encoded string: " << endl;
    cout << BinaryCode << endl;

    DecodeHuffman(BinaryCode, ReducedString, root);
```

```

cout << "Decoded string: " << endl;
cout << ReducedString << endl;

return 0;
}

void Statistics(const char* String, sym simbols[], int& k, int& kk) {
    int kolvo[256] = {0};

    for (int i = 0; i < strlen(String); i++) {
        int chh = String[i];

        for (int j = 0; j < 256; j++) {
            if (chh == simbols[j].ch) {
                kolvo[j]++;
                kk++;
                break;
            }

            if (simbols[j].ch == 0) {
                simbols[j].ch = static_cast<unsigned char>(chh);
                kolvo[j] = 1;
                k++;
                kk++;
                break;
            }
        }
    }

    for (int i = 0; i < k; i++)
        simbols[i].freq = static_cast<float>(kolvo[i]) / kk;

    sym* psym[256];
    for (int i = 0; i < k; i++)
        psym[i] = &simbols[i];

    sym tempp;
    for (int i = 1; i < k; i++)
        for (int j = 0; j < k - 1; j++)
            if (simbols[j].freq < simbols[j + 1].freq) {
                tempp = simbols[j];
                simbols[j] = simbols[j + 1];
                simbols[j + 1] = tempp;
            }
}

```



```

for (int i = 0; i < k; i++) {
    printf("Ch= %d\tFreq= %f\tPPP= %c\t\n", simbols[i].ch,
        simbols[i].freq, psym[i]->ch, i);
}
printf("\n Slova = %d\tSummir=%f\n", kk, static_cast<float>(kk));
}

```

```

sym* makeTree(sym* psym[], int k) {
    int i, j;
    sym* temp;

    temp = new sym;
    temp->freq = psym[k - 1]->freq + psym[k - 2]->freq;
    temp->code[0] = 0;
    temp->left = psym[k - 1];
    temp->right = psym[k - 2];

    if (k == 2)
        return temp;
    else {
        for (i = 0; i < k; i++)
            if (temp->freq > psym[i]->freq) {
                for (j = k - 1; j > i; j--)
                    psym[j] = psym[j - 1];
                psym[i] = temp;
                break;
            }
    }

    return makeTree(psym, k - 1);
}

```

```

void makeCodes(sym* root) {
    if (root->left) {
        strcpy(root->left->code, root->code);
        strcat(root->left->code, "0");
        makeCodes(root->left);
    }

    if (root->right) {
        strcpy(root->right->code, root->code);
        strcat(root->right->code, "1");
        makeCodes(root->right);
    }
}

```

```

void CodeHuffman(const char* String, char* BinaryCode, sym* root) {
    for (int i = 0; i < strlen(String); i++) {
        int chh = String[i];
        for (int j = 0; j < 256; j++)
            if (chh == root->ch) {
                strcat(BinaryCode, root->code);
            }
    }
}

void DecodeHuffman(const char* BinaryCode, char* ReducedString, sym* root) {
    sym* Current = root;
    char CurrentBit;
    int BitNumber = 0;
    int CurrentSimbol = 0;

    while (BitNumber != strlen(BinaryCode)) {
        while (Current->left != nullptr && Current->right != nullptr &&
            BitNumber != strlen(BinaryCode)) {
            CurrentBit = BinaryCode[BitNumber++];
            if (CurrentBit == '0')
                Current = Current->left;
            else
                Current = Current->right;
        }

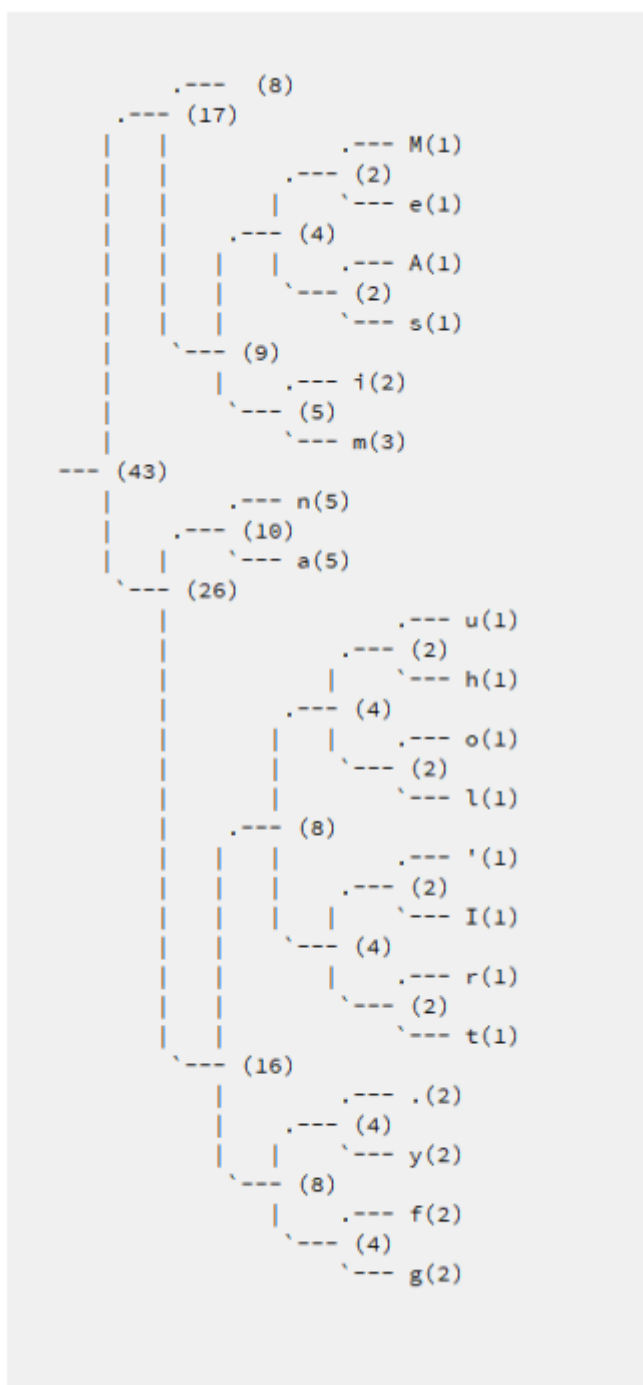
        ReducedString[CurrentSimbol++] = Current->ch;
        Current = root;
    }

    ReducedString[CurrentSimbol] = 0;
}

```

ДОДАТОК Д

Візуалізація дерева Хаффмана



ДОДАТОК Е

Блок-схема алгоритму декодування методом Хаффмана

