**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ   
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп’ютерних наук та кібернетики

Кафедра математичної інформатики

«До захисту допущено»

Завідувач кафедри

Терещенко В.М

(підпис)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_р.

**Дипломна робота**

**На здобуття ступеня бакалавра**

за спеціальністю 122 Комп’ютерні науки

на тему:

**ПЕРЕТВОРЕННЯ БАРРОУЗА-УІЛЛЕРА ТА ЙОГО ЗАСТОСУВАННЯ**

Виконав студент 4-го курсу  
Волохович Ігор Ігорович

(підпис)

Науковий керівник:

професор, доктор фіз.-мат. наук

Анісімов Анатолій Васильович

(підпис)

Засвідчую, що в цій курсовій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент

(підпис)

Київ – 2021

**Реферат**

Обсяг роботи ХХ сторінки, ХХ ілюстрацій, ХХ таблиць, ХХ джерел посилань.

ПЕРЕТВОРЕННЯ БАРРОУЗА-УІЛЛЕРА, МЕТОДИ СТИСНЕННЯ ДАНИХ, АЛГОРИТМ ЛЕМПЕЛЯ–ЗІВА–ВЕЛЧА, АЛГОРИТМ ХАФФМАНА

Об’єктом роботи є дослідження методів стиснення даних, перетворення Барроуза – Віллера[[9]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1) та доцільність його використання у методах стиснення.

Метою роботи є розробка алгоритмів стиснення даних та перетворення Барроуза–Віллера[[9]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1). Модифікація одного з алгоритму та порівняння якість та час стиснення даних.

Методи розроблення: аналіз алгоритмів стискання, кодування методів стиснення, аналіз якості та часу стиснення алгоритмів та їх модифікацій, розробка загальної бібліотеки алгоритмів, розробка прикладів використання алгоритмів, розробка веб–застосунку для демонстрації можливостей бібліотеки.

Інструменти розроблення. Для розробки реалізації було використано інтерактивне середовище розробки JebBrains Rider[[1]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1). Було використано мову C#[[2]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1). Створено .NET Library[[3]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1) та використовуючи ASP.NET Framework [[4]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1) створено веб – застосунок. Розроблено систему логування часу та обрахування відсотку стиснення.

Результат роботи. Було досліджено методи стиснення даних та вже існуючі продукти стиснення на ринку. Розроблено декілька демонстраційних консольних програм та веб-застосунок. Також було запроваджено стандартну загальну бібліотеку методів, яка було викладена в мережу на правах open-source та доступна всім розробникам. Було показано використання бібліотеки на різних платформах.

**Зміст**

[ВСТУП 4](#_Toc70961041)

[РОЗДІЛ 1. ЗАГАЛЬНІ ВІДОМОСТІ ПРО МЕТОДИ СТИСНЕННЯ ДАНИХ 6](#_Toc70961042)

[1.1 Перше застосування терміну «База Даних» 6](#_Toc70961043)

[РОЗДІЛ 2. ЗАГАЛЬНІ ВІДОМОСТІ ПРО ПЕРЕТВОРЕННЯ БАРРОУЗА-УІЛЛЕРА (BWT) 6](#_Toc70961044)

[2.1 Відомості про Entity Framework 6](#_Toc70961045)

[РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ ПЕРЕТВОРЕННЯ BWT 6](#_Toc70961046)

[3.1 Відомості про Entity Framework[7] 6](#_Toc70961047)

[РОЗДІЛ 4. ІМПЛЕМЕНТАЦІЯ МЕТОДІВ СТИСНЕННЯ 6](#_Toc70961048)

[4.1 Стандартний метод LZW 6](#_Toc70961049)

[4.2 Метод LZW з перетворенням Барроуза-Уіллера 6](#_Toc70961050)

[4.3 Метод Хаффмана (Huffman coding) 6](#_Toc70961051)

[РОЗДІЛ 5. ПОРІВНЯННЯ МЕТОДІВ СТИСНЕННЯ 6](#_Toc70961052)

[ВИСНОВКИ 6](#_Toc70961053)

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 6](#_Toc70961054)

[ДОДАТКИ 6](#_Toc70961055)

# ВСТУП

**Оцінка сучасного стану об’єкта дослідження.** На сьогоднішній день, зі зростанням кількості користувачів та кількості даних в мережі інтернет гостро постає питання зберігання даних. Так як будь-яка людина розуміє, що просто зберігати файли не дуже правильно, люди замислюються щодо використання алгоритмів стиснення задля збереження дорогого та цінного місця на серверах, комп’ютерах, мережевих системах. Через це були придумані та розроблені різні алгоритми стиснення даних та допоміжні перетворення для збільшення якості стиснення. Наприклад великій продуктовій компанії Amazon не вигідно зберігати сотні мільйонів даних про користувачів, товари та історію у первісному вигляді. Це займає багато місця. Тому вони використовують алгоритми, які допомагають в такому же об’ємі пам’яті зберігати набагато більше файлів.

**Актуальність роботи та підстави для її виконання.** Застосування, розробка нових та модифікація існуючих алгоритмів стиснення має важливе значення зі збільшенням генерації інформації у сьогоднішній час. Задля раціонального використання пам’яті та попередження надлишкової інформаційної катастрофи, коли не вистачатиме пам’яті. Методи стиснення та модифіковані (наприклад з використанням алгоритму Барроуза–Віллера) допомагають зменшити розмір файлу тим самим зменшуючи витрати та попередження настання інформаційної катастрофи.

**Мета й завдання роботи.** Метою дипломної роботи є огляд алгоритму перетворення Барроуза–Віллера, модифікація алгоритму Лемпеля – Зіва – Велша використовуючи цей алгоритм та порівняльний аналіз результатів стиснення, які приведуть до висновку модифікування алгоритмів у такий спосіб. Також завданням є реалізація власне перетворення BWT(Барроуза – Віллера)[[9]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1), алгоритму стиснення LZW(Лемпеля – Зіва – Велша)[[12]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1), модифікація даного алгоритму використовуючи перетворення та реалізація алгоритму Huffman coding(алгоритм стиснення Хаффмана)[[8]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1). Порівняння даних результатів у розрізі часу виконання та результуючому розмірі файлу. Для досягнення цієї поставлено такі завдання:

* Огляд методу BWT[[9]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1).
* Огляд основних методів та алгоритмів стиснення.
* Провести роботу у дослідженні розробки бібліотек
* Розробити програмну реалізацію алгоритму BWT[[9]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1), LZW[[12]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1), LZW + BWT, Huffman coding[[8]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1) з можливістю використовувати у будь-якому проекті на базі мови C#[[2]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1).

**Об'єкт, методи й засоби розроблення**. Об’єктом порівняльного аналізу є стандартні та модифіковані алгоритми стиснення даних. Об’єктом розробки є декілька програмних реалізацій, які допомагають побачити та застосувати бажаний алгоритм стиснення та бібліотеку з алгоритмами, яка може бути використана та встановлена будь-яким розробником через NuGet[[4]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1) пакет у середовищі мови програмування C#[[2]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1)

# РОЗДІЛ 1. ЗАГАЛЬНІ ВІДОМОСТІ ПРО СТИСНЕННЯ ДАНИХ

## Що таке стиснення даних?

Стиснення даних – це зменшення кількості бітів, необхідних для репродукції даних. Такі алгоритми допомагають зберігати пам’ять, збільшити швидкість передачі даних та зменшити вартість накопичувачів та необхідну пропускну властивість інтернету для передачі. Іншими словами, це процес кодування, реструктуризації або модифікування даних для зменшення її розміру. Фундаментально, це включає повторне шифрування інформації, користуючись меншими бітами (шматками), ніж оригінальне представлення. Стиснення даних підлягає компромісу між кількістю пам’яті, яку займає та часом. Це широко застосовується в обчислювальних сервісах та рішеннях, особливо в комунікації даними, Стискання даних працює за допомогою декількох методів стиснення та програмних рішень, які використовують алгоритми стискання для зменшення обсягу даних.

## Історія створення методів стиснення

Теоретичну базу стискання даних було розроблено Клодом Шеноном, який опублікував роботи по алгоритмічній теорії інформації(AIT) [[13]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1) для компресії без втрат та теорії швидкості спотворення[[14]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1) для компресії з втратами. Вони стали фундаментом започаткування стискання даних, які на сьогоднішній день досягли величезних здібностей та дуже часто застосовуються у машинному навчанні.

З народженням інтернету в 70 – х роках, взаємозв’язок між розміром файлів та швидкістю передачі даних став більш значущим. Видатні розуми всього світу боролися з цією проблемою роками, але безрезультатно допоки не з’явився у середині 80 – х роках універсальний алгоритм стиснення без втрат Лемпеля – Зіва – Велша (LZW)[[12]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1), який показав реальні переваги стискання.

Метод LZW був першим широко вживаним алгоритмом стискання даних, який було розроблено на комп’ютерах та досі використовується сьогодні (у різноманітних варіаціях). Великий текст англійською мовою може бути стиснутий приблизно на 50% використовуючи LZW[[6]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1). Але історія методів починає задовго до цього. Азбука Морзе, яка була створена у 1838 році, являється предком такого поняття як «стискання даних». Тобто, це є одним із найперших методів стискання. Таким алгоритмом, в якому найпоширеніші літери англійської мові, такі як «е» і «t» мають коротші азбуки Морзе.

Пізніше, в 1949 році, коли почали освоюватися мейнфрейми, вже відомий нам Клод Шеннон разом з Робертом Фано розробили кодування Шеннона – Фано[[11]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1). Їх алгоритм призначав коди символам у заданих блоках базуючись на ймовірності появи символу. Ймовірність появи символу обернено пропорційна довжині коду, що призводить до більш короткого способу презентації даних[[5]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1).

Два роки потому, Девід Хаффман вивчав курс «Теорія Інформації» в Массачусетському Технологічному Інституті (МІТ) та мав заняття, викладачем яких був Робертом Фано. Викладач запропонував класу вибір: написати курсову робочу або складати випускний іспит. Авжеж Хаффман виправ курсову, тема якої була «Пошук найбільш ефективного методу двійкового кодування». Попрацювавши кілька місяців та нічого не придумавши, Хаффман вже збирався викинути всю свою роботу й почати готуватися до випускного іспиту замість здачі курсової. Саме в цей час йому в голову прийшла ідея, просвітництво та він придумав дещо схожу на техніку Шеннона – Фано[[11]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1), але набагато ефективнішу. Ключова відмінність між кодуванням Хаффмана та зазначеним вище було те, що у алгоритмі Шеннона – Фано[[11]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1) дерево ймовірностей будувалося знизу вверх, створюючи неоптимальний результат, а в цьому алгоритмі – зверху вниз.

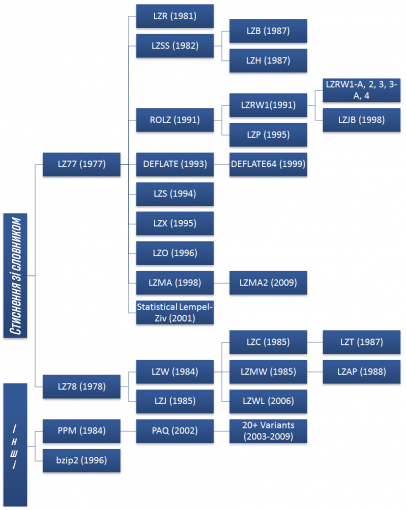


Рисунок 1. Ієрархія алгоритмів стиснення «без втрат»

Ранні імплементації алгоритмів Хаффмана[[8]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1) та Шеннона – Фано[[11]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1) були розроблені з використанням апаратних та жорстко закодованих (hard coded) кодів. Але коди Хаффмана до цього були статичними. Завдяки появі Інтернету в 1970 – х роках та сервісів, які надаюсь послуги онлайн – сховищ, розробники змогли вдосконалити алгоритм та реалізувати такий застосунок, який генерував коди Хаффмана динамічно застосовуючи вхідні дані як основу. Пізніше, в 1977 році, Авраам Лемпель та Якоб Зів опублікували свою інноваційний, новаторський алгоритм LZ77[[6]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1). Це був перший алгоритм, який для стиснення використовує словник. Конкретніше, LZ77[[6]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1) часто використовував динамічний словник, який називається «ковзаючим вікном» (sliding window)[[10]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1). У 1978 році цей же дует опублікував новий алгоритм LZ78[[7]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1), який теж використовує словник. На відміну від LZ77[[6]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ_1), цей алгоритм аналізує вхідні дані та генерує статичний словник, а не генерує його динамічно.

## Зростання дефляції

З моменту виходу алгоритмів дуету Лемпеля – Зіва, компанії та інші великі організації використовують алгоритми стиснення даних, оскільки їхні потреби в зберіганні та сховищі продовжують зростати і стиснення даних може повністю задовольнити їх потреби.

## Типи стиснення

## Переваги та недоліки стискання даних

# РОЗДІЛ 2. ОГЛЯД МЕТОДІВ СТИСНЕННЯ

## Відомості про Entity Framework

# РОЗДІЛ 3. ОГЛЯД АЛГОРИТМІВ СТИСНЕННЯ

# РОЗДІЛ 4. ІМПЛЕМЕНТАЦІЯ ПЕРЕТВОРЕННЯ BWT

## Відомості про Entity Framework[[7]](#_ПЕРЕЛІК_ВИКОРИСТАНИХ_ДЖЕРЕЛ)

# РОЗДІЛ 4. ІМПЛЕМЕНТАЦІЯ МЕТОДІВ СТИСНЕННЯ

## Стандартний метод LZW

## Метод LZW з перетворенням Барроуза-Віллера

## Метод Хаффмана (Huffman coding)

## РОЗДІЛ 5. ПОРІВНЯННЯ РЕЗУЛЬТАТІВ

# ВИСНОВКИ

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jeb Brains Rider [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.jetbrains.com/rider/>.
2. C# programming language [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/C_Sharp_(programming_language)>.
3. .NET Library [Електронний ресурс] // Microsoft. – 2018. – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/standard/library-guidance/>.
4. ASP.NET Framework [Електронний ресурс] // Microsoft. – 2021. – Режим доступу до ресурсу: <https://dotnet.microsoft.com/apps/aspnet>.
5. Стівен В. Новий вид Науки / Вольфрам Стівен. – Іллінойс: Вольфрам Медіа, 2002. – 1069 с.
6. Lossless Data Compression: LZ77 [Електронний ресурс] – Режим доступу до ресурсу: <https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossless/lz77/index.htm>.
7. Lossless Data Compression: LZ78 [Електронний ресурс] – Режим доступу до ресурсу: <https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossless/lz78/index.htm>.
8. Huffman encoding [Електронний ресурс] – Режим доступу до ресурсу: <http://homes.sice.indiana.edu/yye/lab/teaching/spring2014-C343/huffman.php>.
9. Burrows-Wheeler Transform [Електронний ресурс] – Режим доступу до ресурсу: <https://web.stanford.edu/class/cs262/presentations/lecture4.pdf>.
10. Sliding Window Algorithm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.baeldung.com/cs/sliding-window-algorithm>.
11. Lossless Data Compression: Shannon-Fano [Електронний ресурс] – Режим доступу до ресурсу: <https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossless/shannonfano/index.htm>.
12. LZW Encoding [Електронний ресурс] – Режим доступу до ресурсу: <http://www.cs.columbia.edu/~allen/S14/NOTES/lzw.pdf>.
13. Grunwald P. D. Algorithmic Information Theory: дис. докт. філос. наук / Grunwald Peter D. – Амстердам, 2007. – 37 с.
14. Rate-Distortion Theory [Електронний ресурс] – Режим доступу до ресурсу: https://web.stanford.edu/class/ee368b/Handouts/04-RateDistortionTheory.pdf.

# ДОДАТКИ

ДОДАТОК А