

# Лабораторна робота № 2

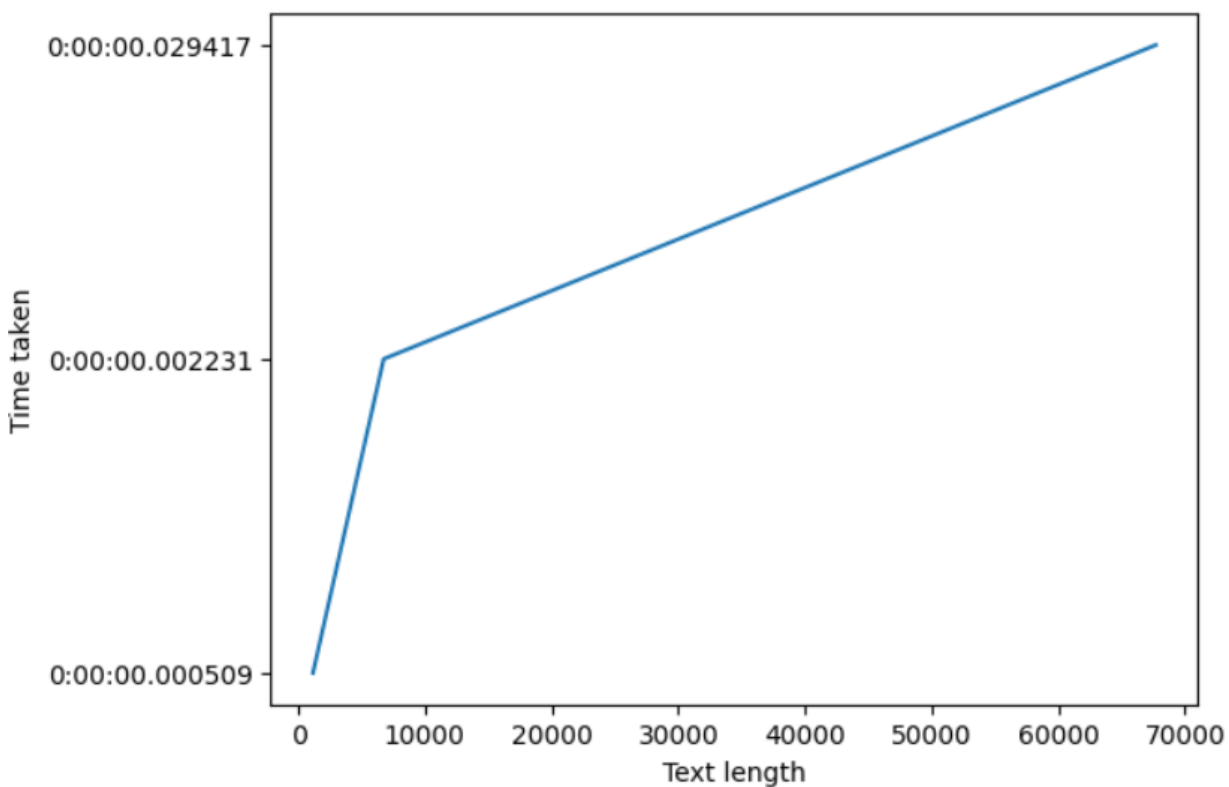
## ❖ Розподіл роботи:

Грищенко Андрій: LZ77, Deflate, оформлював у ірυνЬ та робив тестування і порівняння

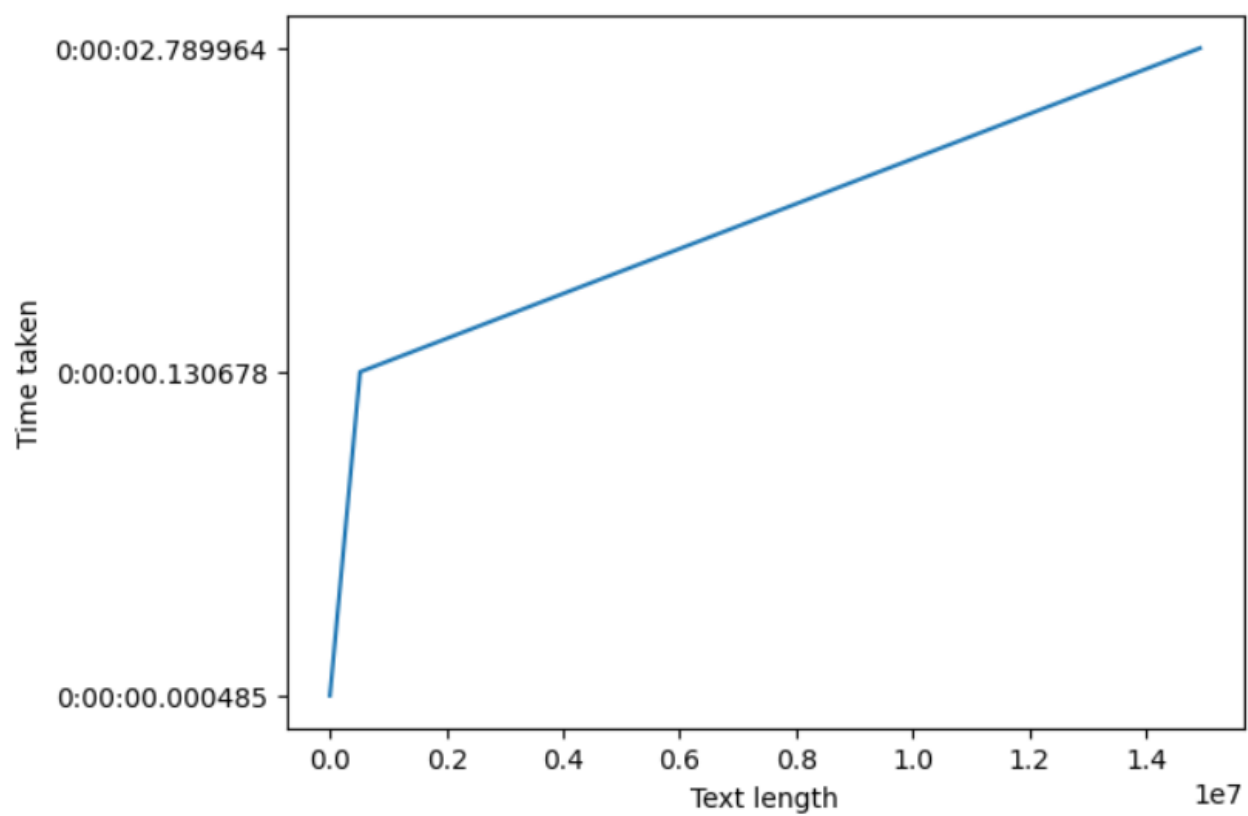
Савчук Оксана: LZW, Huffman, писала звіт

## ❖ Графіки часу роботи відносно розміру вхідних даних для кожного з алгоритмів

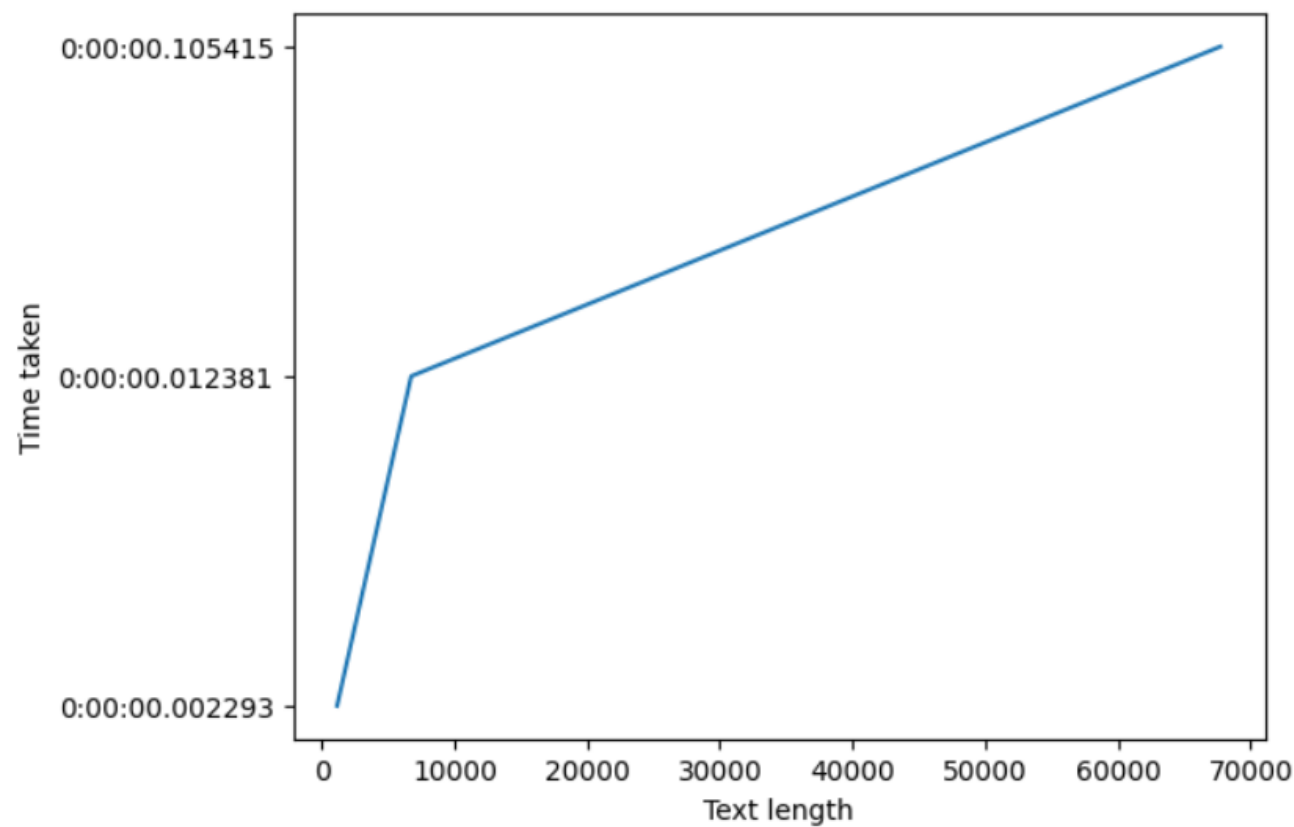
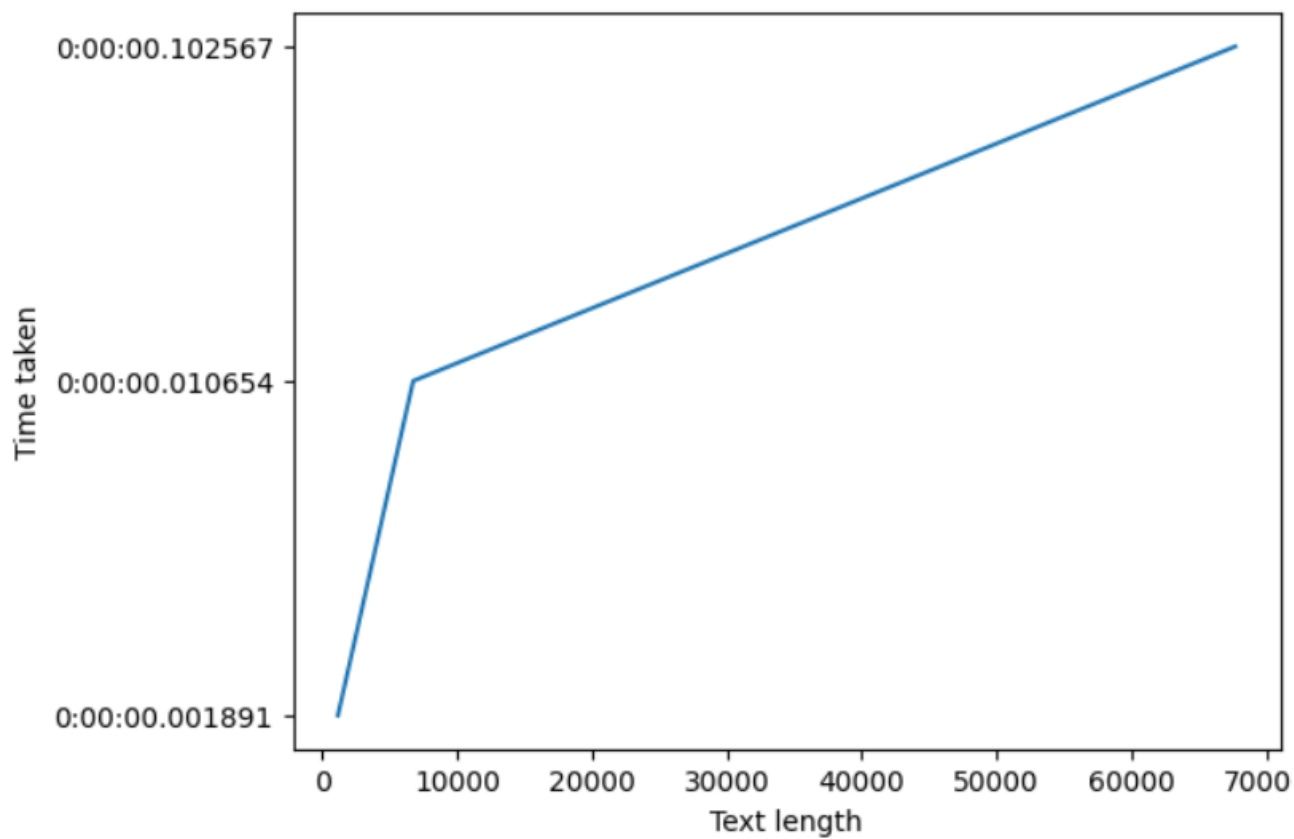
### ★ LZW алгоритм

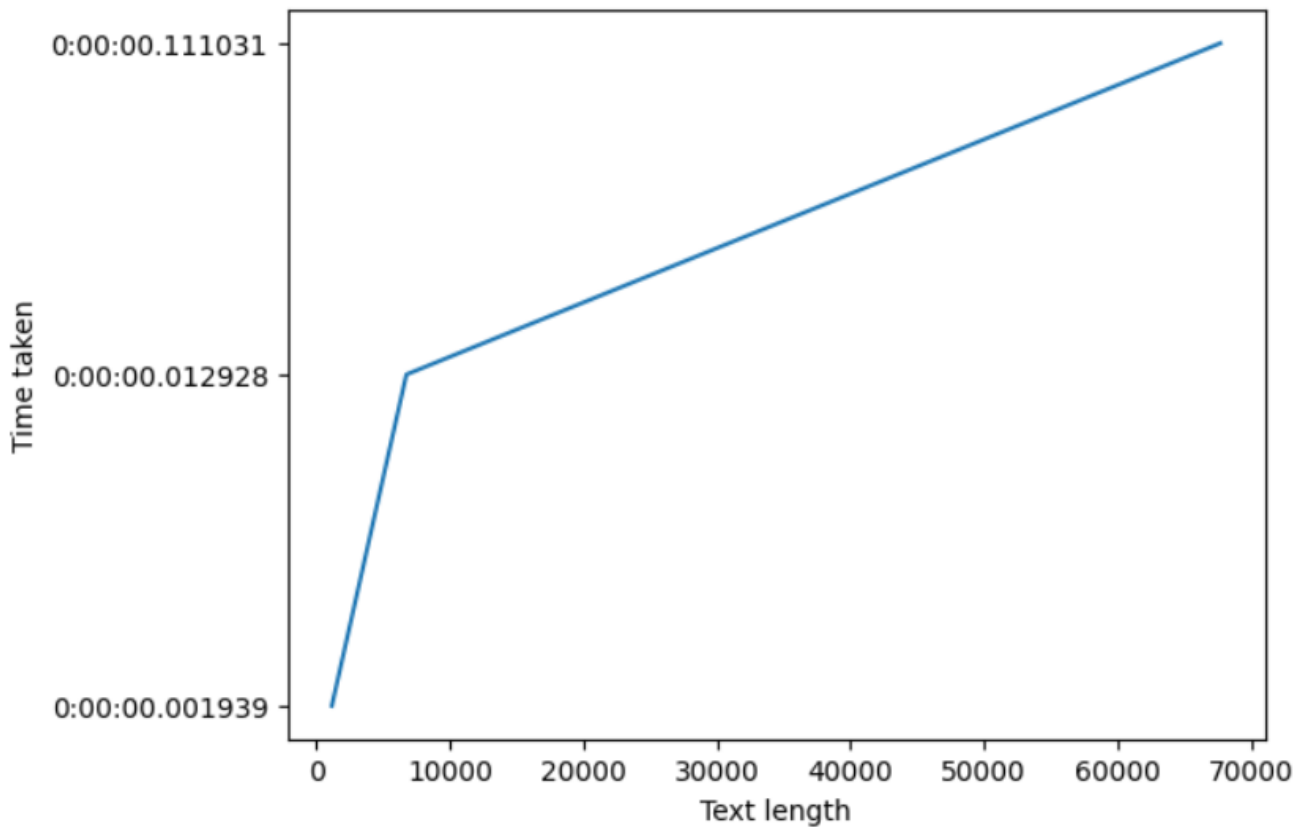


### ★ Алгоритм Хаффмана



★ LZ77 алгоритм





### ❖ Оцінити ступінь стиснення відносно розміру вхідних даних (у відсотках)

#### ★ LZW та LZ77 алгоритми

Найвищий ступінь стиснення(99%) є при найбільшій кількості повторювальних даних.

#### ★ Алгоритм Хаффмана

При великій кількості повторювальних даних ступінь стиснення буде найнижчим(47-72%)

### ❖ Короткий висновок для кожного із алгоритмів про те, коли краще його використовувати та за яких умов він дає кращі результати

#### ★ LZW алгоритм

Цей алгоритм має кращі показники стиснення, коли ми працюємо з текстовими даними, особливо якщо вони містять багато повторень.

#### ★ LZ77 алгоритм

Цей алгоритм дає кращі результати стиснення для даних, які містять багато повторень. Але якщо дані не містять багато повторень, то алгоритм може бути менш ефективним.

#### ★ Алгоритм Хаффмана

Цей алгоритм дає кращі результати для даних з нерівномірною частотою входження символів. Також він менш ефективний для даних, що містять багато повторюваних рядків.

### ❖ **Загальний висновок, у якому для певних умов навести алгоритм, що поводитиметься найкраще (багато даних, що повторюються тощо)**

Якщо дані для стиснення є текстовими та з повтореннями, то краще використовувати LZW та LZ77. Якщо ж ви хочете стиснути дані з нерівномірною частотою входження символів, то краще обрати Алгоритм Хаффмана.

Можна зробити висновок, що при виборі найкращого алгоритму стиснення потрібно врахувати конкретні вимоги(пр: повторювальність даних).