

## Звіт

### Лабораторна робота з дискретної математики

*Виконали Агітольєв Андрій та Озарко Богдан*

#### 1. Розподіл роботи між учасниками:

Андрій:

1. Написання алгоритму Гаффмана
2. Аналіз ефективності алгоритмів в залежності від вхідних даних
3. Оформлення файлу «Jupyter Notebook»

Богдан:

1. Написання алгоритмів Lzw та Lz77
2. Оцінка ступеня стиснення
3. Оформлення звіту

#### 2. Час роботи в залежності від розміру вхідних даних:

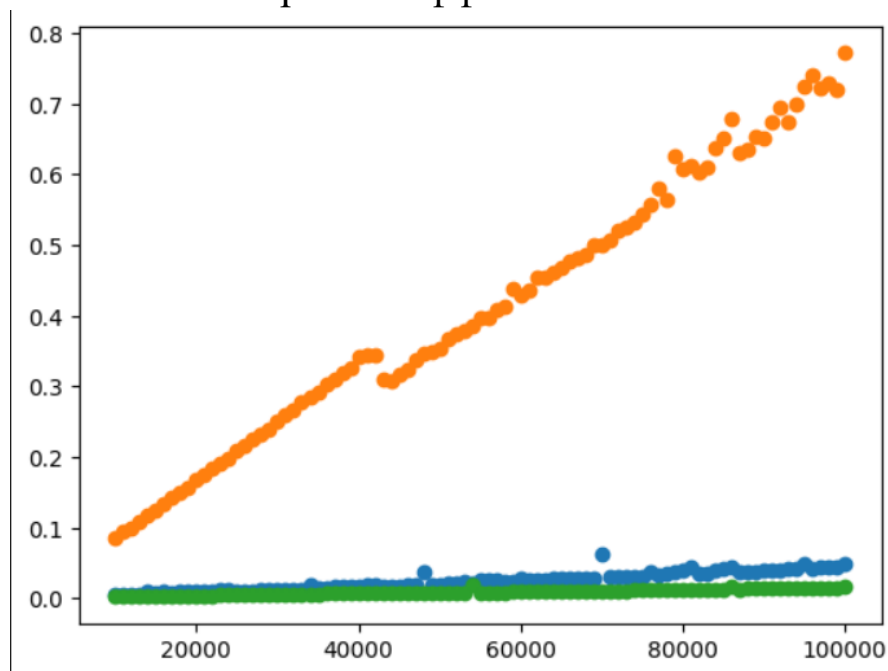
Розмір буферу для lz77 в прикладі – 100

В якості даних були взяті випадково згенеровані рядки на проміжку (10000 – 100000 символів з кроком 1000)

Синій графік – lzw

Оранжевий – lz77

Зелений – алгоритм Гаффмана



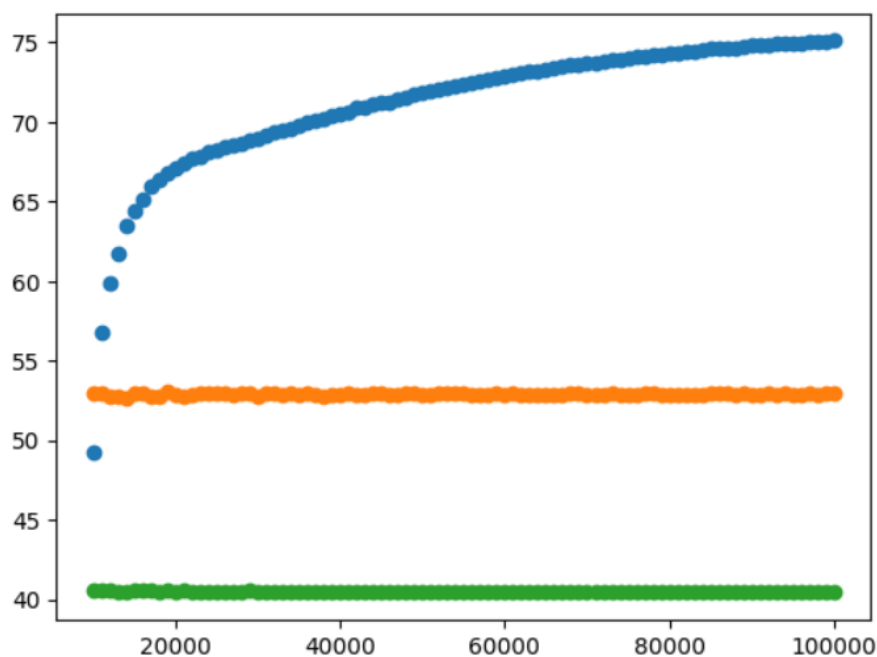
Аналіз:

Lz77 – працює найдовше і показує себе як найменш ефективний, оскільки його набір даних для кодування обмежується буфером, який крім того своїм розміром впливає на час роботи (більший буфер – довше працює). Також сам по собі доволі складний.

Lzw – значно швидший за Lz77, проте трохи повільніший за Гаффмана, кодує значно ефективніше, оскільки його словник не обмежується зверху і може постійно поповнюватися новими патернами.

Гаффман – оскільки базується на повторюваності кожного символу, то його ефективність залежить вибірки символів у даних. Це робить його дуже ефективним для тексту, де вибірка може обмежитись одним алфавітом.

### 3. Оцінка ступеня стиснення (на тих самих даних)



Аналіз:

Lz77 – стискує ліпше за Гаффмана попри довгий час роботи і робитиме це ще краще за умови великого буфера. Оскільки він дивиться на повторювані рядки, то копіює їх, що записує одразу кілька символів в один код

Lzw – робить процес стиснення найкраще, оскільки має великий словник і простий код, що робить його максимально коротким порівняно з іншими алгоритмами. Чим більший розмір даних, тим

багатший словник у Lzw і тим простіший код. Працює швидше від Lz77, бо його словник не обмежений буфером і процес кодування простіший

Гаффман – попри те, що працює найшвидше, стикає найгірше, оскільки спирається на кожен символ індивідуально та не дивиться на повтори. При великій вибірці символів стискатиме особливо погано.

#### 4. Висновки:

Обидва алгоритми, Гаффмана і Lzw є дуже швидкими та ефективними, а тому і популярними алгоритмами стиску. Вони принципово різняться в підході до кодування вхідних даних. Метод Гаффмана потребує першочергово дані про повторюваність кожного символу, який потім кодує в двійковому вигляді. LZW ж цього не потребує, і формує в результаті роботи словник з даними паттернів (послідовностей символів). Зазвичай Гаффман швидший за LZW, але й він може робити кодування швидше на окремих типах даних, де паттерни повторюються дуже часто. Щодо LZ77, він стискає все ще ліпше за Гаффмана, бо дивиться на повторюваність, але гірший за LZW, бо має обмежений буфер. Крім цього повільний, бо процес кодування доволі складний.

Отже, Гаффман ефективний для малої вибірки символів, проте погано стискає, Lz77 стискає в рази краще за рахунок орієнтації на повторювані символи, проте кодує довго, Lzw є поліпшеною версією Lz77, оскільки необмежений буфером та працює швидше.