Лабораторна робота №1

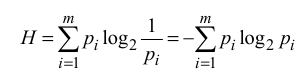
Дослідження кількості інформації при різних варіантах кодування

**Мета:** Дослідити імовірнісні параметри української мови для оцінки кількості інформації текстів. Дослідити вплив різних методів кодування інформації на її кількість.

# Теоретичні відомості

***Відносна частота появи символу*** - імовірність появи певного символу в певному місці тексту - відношення числа появи символу в тексті до загальної кількості символів.

***Середня ентропія нерівноймовірного алфавіту***:



де *m* - кількість символів алфавіту, *p* - імовірність появи символу

Ентропія вимірюється в **БІТАХ** (як представлення кількості можливих варіантів).

***Кількість інформації в тексті*** - середня ентропія вихідного алфавіту помножена на кількість символів тексту. (***HINT***: результат обрахунку для порівняння значення з розміром файлів треба перевести з бітів в байти)

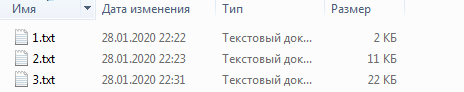
<https://github.com/comradeFreeman/csc.git> ( директорія lab1 )

директорія «archives» містить всі архіви, які були створені під час виконання лаби.

task3v2.py – завдання 3: частоти символів, ентропія, кількість інформації

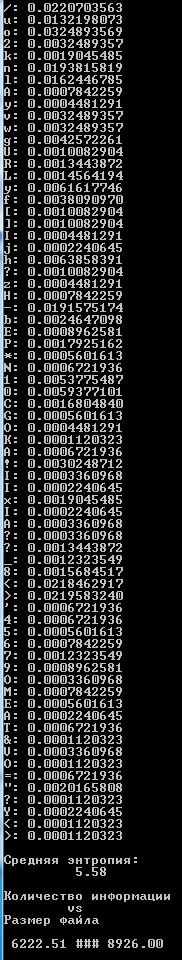
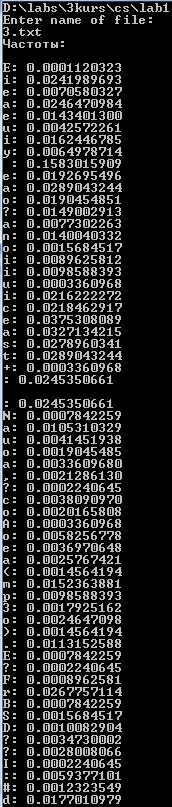
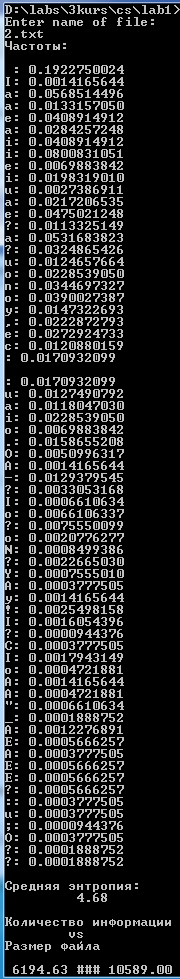
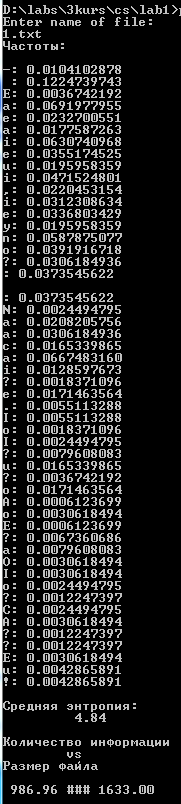
base64en.py – кодування вхідного файлу у base64 та запис в окремий файл

# 1. Дослідження кількості інформації в тексті

1. 

2. Наврядчи хтось ще візьме «Балладу о прокуренном вагоне» О. Кочеткова, «Улыбку» Р. Бредбері та статтю «Локальная инет радиостанция при помощи icecast +ices» з Яндекс.Дзену :)

3.



4. Проведіть стиснення кожного вхід. файлу за допомогою 5 різних алгоритмів

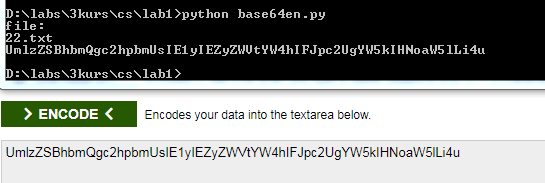
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ориг. файл | Размер | кол-во информации | rar | zip | gz | bz2 | xz |
| 1.txt | 1633 | 986.96 | 822 | 850 | 730 | 784 | 836 |
| 2.txt | 10589 | 6194.63 | 5535 | 5344 | 5224 | 4889 | 5356 |
| 3.txt | 8926 | 6222.51 | 3892 | 3892 | 3772 | 3619 | 3684 |

5. У всіх трьох випадках кількість інформації була меншою розміру файлу, бо сам файл окрім даних містить інші службові дані та атрибути. Також вона завжди була більшою за розміри архівів. Це і логічно, бо використовуються досконалі алгоритми стискання інформації. Також було помічено, що файл, де ентропія більша, вдається стиснути на більший відсоток відносно кількості інформації, відсоток ж відносно розміру файлу на диво сталий і складає близько 40-50%.

Кількість інформації при стисненні зменшилася, але значно зріс показник ентропії та відсоток кількості інформації до розміру файлу (прибл. 98-99%), тобто інформація є більш «щільною» (?)

# 2. Дослідження способів кодування інформації на прикладі Base64

1. Ознайомтесь зі стандартом [RFC4648](https://tools.ietf.org/html/rfc4648)
2. Для практичного засвоєння методу кодування, створіть програму, що кодує довільний файл в Base64 (шляхом реалізації алгоритму вручну, а не виклику бібліотечної функції)
   1. перевірте коректність роботи програми, порівнявши результат з існуючими програмними засобами (наприклад, openssl enc -base64)



1. Закодуйте в Base64 обрані вами текстові файли
   1. Обрахуйте кількість інформації в base64-закодованому варіанті файлу
   2. Порівняйте отримане значення з кількістю інформації вихідного файлу
   3. Зробіть висновки з отриманого результату
2. Закодуйте в Base64 стиснені кращим з алгоритмів текстові файли
   1. Обрахуйте кількість інформації в base64-закодованому варіанті стисненого файлу
   2. Порівняйте отримане значення з кількістю інформації вихідного файлу та base64-закодованого файлу[[1]](#footnote-1)
   3. Зробіть висновки з отриманого результату

|  |  |  |  |
| --- | --- | --- | --- |
| Файл | Количество информации (размер файла)  (энтропия) | | |
| Оригинал | base64 | base64 + bz2 |
| 1.txt || base64\_1\_txt.txt | 986.96 (1633)  (4.84) | 2261.65 (3752)  (4.82) | 1186.07 (1216)  (7.80) |
| 2.txt || base64\_1\_txt.txt | 6194.63 (10589)  (4.68) | 14395.15 (23992) (4.80) | 6094.94 (6124)  (7.96) |
| 3.txt || base64\_3\_txt.txt | 6222.51 (8926)  (5.58) | 10218.20 (14908) (5.48) | 5314.77 (5342)  (7.96) |

Як видно з таблиці та діаграми, при кодуванні у base64 показник ентропії майже не змінився, а якраз кількість інформації суттєво збільшилася (мені здається, так сталося через кодування, яке використовує алфавіт невеликої (точно меншої за вихідний) довжини, 64 символи). Але і розмір файлу теж збільшився пропорційно. При використанні алгоритмів стискання розміри файлів стали відчутно менші навіть оригінальних, а показник ентропії збільшився майже в два рази, що не могло сказатися на кількості інформації, відсоток якої відносно загального розміру файлу сягнув

97-99% (при 60-68% за кодування base64 та приблизно стільки ж у вихідних файлах). На мою думку, це пов’язано з більшою довжиною алфавіту, що використовувався в кодуванні, завдяки чому вдалося досягти найбільшої «щільності»(?) інформації. Також свою роль зіграли самі алгоритми стискання, які зробили зберігання найефективнішим.

1. [↑](#footnote-ref-1)