Домашнее задание 3

Весь код архиватора лежит тут: https://github.com/Andrew-Zlobin/CMDC/tree/main/task3

Смотреть на результаты будем на примере теста из предыдущего задания. Поэтому, для начала откроем его:

In [1]:
```
text = None
with open('war_peace_ascii_Zlobin_AS.txt', 'r', encoding='utf-8') as file
# with open('test.txt', 'r', encoding='utf-8') as file:
    text = file.read()
```

In [2]:
```
text[:1000]
```

Out[2]:
```
'\nCHAPTER I\n\n"Well, Prince, so Genoa and Lucca are now just family es
tates of the\nBuonapartes. But I warn you, if you don\'t tell me that th
is means war,\nif you still try to defend the infamies and horrors perpe
trated by that\nAntichrist-I really believe he is Antichrist-I will have
 nothing\nmore to do with you and you are no longer my friend, no longer
 my\n\'faithful slave,\' as you call yourself! But how do you do? I see I
 \nhave frightened you-sit down and tell me all the news."\n\nIt was in J
uly, 1805, and the speaker was the well-known Anna Pavlovna\nScherer, ma
id of honor and favorite of the Empress Marya Fedorovna.\nWith these wor
ds she greeted Prince Vasili Kuragin, a man of high\nrank and importance
, who was the first to arrive at her reception. Anna\nPavlovna had had a
cough for some days. She was, as she said, suffering\nfrom la grippe; gr
ippe being then a new word in St. Petersburg, used\nonly by the elite.\n
\nAll her invitations without exception, written in French, and delivere
d\nby a scarle'
```

В файлах https://github.com/Andrew-Zlobin/CMDC/blob/main/task3/BWT.py и https://github.com/Andrew-Zlobin/CMDC/blob/main/task3/DC.py реализованы алгориты BWT и DC соответсвенно.

In [3]:
```
from BWT import BWT
from DC import DC
from utils import BWT_DC_encode_pipeline, BWT_DC_decode_pipeline, alphabe
```

Напишем вспомогательные функции, чтобы убедиться, что они работают корректно:

In [4]:
```
def BWT_DC_encode(text):
    bwt_text = BWT.forward(text)
    alphabet = "\x01" + "".join(sorted(list(set(text))))
    int_alphabet = alphabet_to_number(alphabet)
    print("end symbol in bwt", '\x01' in bwt_text)
    print("end symbol in alphabet", '\x01' in alphabet)
    print("bwt_text = ", bwt_text)
    dc_text = DC.code(bwt_text, alphabet, BWT.get_char_spacing())
    text_len = len(bwt_text)
    array_to_encode = [text_len] + dc_text
    return array_to_encode, int_alphabet
```

```
def BWT_DC_decode(array_to_encode, int_alphabet):
    text_len = array_to_encode[0]
    dc_text = array_to_encode[1:]
    alphabet = number_to_alphabet(int_alphabet)
    dc_decoded = DC.decode(dc_text, alphabet, text_len)
    bwt_decoded = BWT.reverse(dc_decoded)
    print("end symbol in dc", '\x01' in bwt_decoded)
    return bwt_decoded
```

In [5]:
```
prepared_list, alph = BWT_DC_encode(text)
```

```
/home/dr_drew/Projects/CMDC_env/lib/python3.11/site-packages/pydivsufsort/
divsufsort.py:103: UserWarning: converting str argument uses more memory
  inp_p = _get_bytes_pointer(inp)
100%|
```

```
| 320165
0/3201650 [00:00<00:00, 5721997.79it/s]
IOPub data rate exceeded.
The Jupyter server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--ServerApp.iopub_data_rate_limit`.

Current values:
ServerApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
ServerApp.rate_limit_window=3.0 (secs)
```

In [6]:
```
decoded_text = BWT_DC_decode(prepared_list, alph)
```

```
end symbol in dc False
```

In [7]:
```
decoded_text[:100]
```

Out[7]:
```
'\nCHAPTER I\n\n"Well, Prince, so Genoa and Lucca are now just family es
tates of the\nBuonapartes. But I '
```

Строки совпадают, значит алгоритмы работают верно

In [8]:
```
[i for i, j in zip(text, decoded_text) if i != j]
```

Out[8]:
```
[]
```

Посмотрим на небольшую статистику массива, который получается после DC:

In [9]:
```
max(prepared_list[1:]), max(prepared_list)
```

Out[9]:
```
(3007480, 3201650)
```

In [10]:
```
len(text)
```

Out[10]:
```
3201649
```

In [11]:
```
len(prepared_list)
```

Out[11]:
```
1378857
```

In [12]:
```
sum([1 for el in prepared_list if el <= 254])
```

Out[12]: 1343183

In [13]:
```
sum([1 for el in prepared_list if el > 254 and el <= 65789])
```

Out[13]: 35279

In [14]:
```
sum([1 for el in prepared_list if el > 254 and el > 65789])
```

Out[14]: 395

Итого, получается, что текст длиной 3201649 символов, преобразуется в массив из 1378857 чисел, 1343183 меньше 254, 395 больше 65789 и 35279 лежат между 254 и 65789

Для сжатия этого массива будем использовать дельта-код Элиаса, арифметическое, и кодирование с переполнением Они реализованы соответственно в:
https://github.com/Andrew-Zlobin/CMDC/blob/main/task3/compression/elias.py
https://github.com/Andrew-Zlobin/CMDC/blob/main/task3/compression/arithmetic.py
https://github.com/Andrew-Zlobin/CMDC/blob/main/task3/compression/overflow.py

И резульаты работы на тексте из предыдущего задания

Дельта код:

Кодирование

In [19]:
```
%%time
!python3 compressor.py war_peace_ascii_Zlobin_AS.txt -d -f result.compres
```

```
/home/dr_drew/Projects/CMDC_env/lib/python3.11/site-packages/pydivsufsort/divsufsort.py:103: UserWarning: converting str argument uses more memory
  inp_p = _get_bytes_pointer(inp)
100%|████████████████████████████| 3201650/3201650 [00:00<00:00, 5813840.89it/s]
CPU times: user 20.4 ms, sys: 12.2 ms, total: 32.6 ms
Wall time: 2.74 s
```

Размер закодированного файла:

In [20]:
```
!ls -l result.compressed
```

```
-rw-r--r-- 1 dr_drew dr_drew 919189 июн 23 17:50 result.compressed
```

Декодирование:

In [ ]:
```
%%time
!python3 compressor.py result.compressed -d -f decoded.txt
```

CPU times: user 2.61 s, sys: 580 ms, total: 3.19 s Wall time: 4min 1s

Проверяем, чтобы файл совпал с исходным:

In [22]:
```
!cmp war_peace_ascii_Zlobin_AS.txt decoded.txt
```

Арифметическое кодирование

Кодирование

In [23]:
```
%%time
!python3 compressor.py war_peace_ascii_Zlobin_AS.txt -a -f result.compres
```

```
/home/dr_drew/Projects/CMDC_env/lib/python3.11/site-packages/pydivsufsort/
divsufsort.py:103: UserWarning: converting str argument uses more memory
  inp_p = _get_bytes_pointer(inp)
100%|████████████████████████████| 3201650/3201650 [00:00<00:00, 5822550.3
4it/s]
CPU times: user 988 ms, sys: 193 ms, total: 1.18 s
Wall time: 3min 40s
```

Размер закодированного файла:

In [24]:
```
!ls -l result.compressed
```

```
-rw-r--r-- 1 dr_drew dr_drew 863866 июн 23 17:57 result.compressed
```

Декодирование:

In [25]:
```
%%time
!python3 compressor.py result.compressed -a -f decoded.txt
```

```
CPU times: user 1.05 s, sys: 320 ms, total: 1.37 s
Wall time: 4min 15s
```

Проверяем, чтобы файл совпал с исходным:

In [26]:
```
!cmp war_peace_ascii_Zlobin_AS.txt decoded.txt
```

Кодирование с переполнением:

Кодирование

In [27]:
```
%%time
!python3 compressor.py war_peace_ascii_Zlobin_AS.txt -o -f result.compres
```

```
/home/dr_drew/Projects/CMDC_env/lib/python3.11/site-packages/pydivsufsort/
divsufsort.py:103: UserWarning: converting str argument uses more memory
  inp_p = _get_bytes_pointer(inp)
100%|████████████████████████████| 3201650/3201650 [00:00<00:00, 5819734.2
4it/s]
CPU times: user 18.7 ms, sys: 5.48 ms, total: 24.2 ms
Wall time: 2.46 s
```

Размер закодированного файла:

In [28]:
```
!ls -l result.compressed
```

```
-rw-r--r-- 1 dr_drew dr_drew 1451817 июн 23 18:02 result.compressed
```

Декодирование:

In [29]:
```
%%time
!python3 compressor.py result.compressed -o -f decoded.txt
```

```
CPU times: user 34.4 ms, sys: 24.7 ms, total: 59.1 ms
Wall time: 8.87 s
```

Проверяем, чтобы файл совпал с исходным:

In [30]:
```
!cmp war_peace_ascii_Zlobin_AS.txt decoded.txt
```

In [31]:
```
!ls -l war_peace_ascii_Zlobin_AS.txt
```

```
-rw-r--r-- 1 dr_drew dr_drew 3201649 июн 20 15:02 war_peace_ascii_Zlobin_A
S.txt
```

Итого получилось, что исходный текст объёмом 3.05 мб удалось сжать до 1.38 мб кодированием с переполнением, до 0.87 мб дельта кодом и до 0.82 мб арифметическим кодированием. Арифметическое кодирование показало наилучший результат

In [ ]:
```
```