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

код Фибоначчи

Кодирование

In [43]:
```
%%time
!python3 compressor.py war_peace_ascii_Zlobin_AS.txt -i -f result_fib.com
```

```
/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, 5914465.5
1it/s]
compressed to 911671 bytes, (or 0.869 MB)
CPU times: user 24.1 ms, sys: 8.54 ms, total: 32.6 ms
Wall time: 3.38 s
```

Размер закодированного файла:

In [44]:
```
!ls -l result_fib.compressed
```

```
-rw-r--r-- 1 dr_drew dr_drew 911671 июн 24 00:41 result_fib.compressed
```

Декодирование:

In [45]:
```
%%time
!python3 compressor.py result_fib.compressed -i -f decoded_fib.txt
```

```
CPU times: user 49.1 ms, sys: 18.2 ms, total: 67.2 ms
Wall time: 9.23 s
```

Проверяем, чтобы файл совпал с исходным:

In [46]:
```
!cmp war_peace_ascii_Zlobin_AS.txt decoded_fib.txt
```

Дельта код:

Кодирование

In [47]:
```
%%time
!python3 compressor.py war_peace_ascii_Zlobin_AS.txt -d -f result_delta.c
```

```
/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, 5853325.4
5it/s]
compressed to 919189 bytes, (or 0.877 MB)
CPU times: user 21 ms, sys: 8.89 ms, total: 29.9 ms
Wall time: 2.78 s
```

Размер закодированного файла:

In [48]:
```
!ls -l result_delta.compressed
```

```
-rw-r--r-- 1 dr_drew dr_drew 919189 июн 24 00:44 result_delta.compressed
```

Декодирование:

In [49]:
```
%%time
!python3 compressor.py result_delta.compressed -d -f decoded_delta.txt
```

```
CPU times: user 1.4 s, sys: 274 ms, total: 1.67 s
Wall time: 4min 7s
```

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

Проверяем, чтобы файл совпал с исходным:

In [50]:
```
!cmp war_peace_ascii_Zlobin_AS.txt decoded_delta.txt
```

Арифметическое кодирование (32 бита)

Кодирование

In [51]:
```
%%time
!python3 compressor.py war_peace_ascii_Zlobin_AS.txt -a -f result_ar.comp
```

```
/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, 5906738.9
6it/s]
compressed to 863866 bytes, (or 0.824 MB)
CPU times: user 997 ms, sys: 242 ms, total: 1.24 s
Wall time: 3min 49s
```

Размер закодированного файла:

In [52]:
```
!ls -l result_ar.compressed
```

```
-rw-r--r-- 1 dr_drew dr_drew 863866 июн 24 00:52 result_ar.compressed
```

Декодирование:

In [53]:
```
%%time
!python3 compressor.py result_ar.compressed -a -f decoded_ar.txt
```

```
CPU times: user 1.11 s, sys: 231 ms, total: 1.35 s
Wall time: 4min 14s
```

Проверяем, чтобы файл совпал с исходным:

In [54]:
```
!cmp war_peace_ascii_Zlobin_AS.txt decoded_ar.txt
```

Кодирование с переполнением:

Кодирование

In [60]:
```
%%time
!python3 compressor.py war_peace_ascii_Zlobin_AS.txt -o -f result_ov.comp
```

```
/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, 5659684.3
3it/s]
compressed to 1451817 bytes, (or 1.385 MB)
CPU times: user 21.1 ms, sys: 7.84 ms, total: 29 ms
Wall time: 2.46 s
```

Размер закодированного файла:

In [61]:
```
!ls -l result_ov.compressed
```

```
-rw-r--r-- 1 dr_drew dr_drew 1451817 июн 24 00:59 result_ov.compressed
```

Декодирование:

In [62]:
```
%%time
!python3 compressor.py result_ov.compressed -o -f decoded_ov.txt
```

```
CPU times: user 56.2 ms, sys: 9.58 ms, total: 65.8 ms
Wall time: 8.7 s
```

Проверяем, чтобы файл совпал с исходным:

In [63]:
```
!cmp war_peace_ascii_Zlobin_AS.txt decoded_ov.txt
```

In [64]:
```
!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
```

Кодирование с переполнением, а потом арифметическое

In [78]:
```
%%time
!python3 compressor.py war_peace_ascii_Zlobin_AS.txt -c -f result_ov_ar.c
```

```
/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, 5926114.2
5it/s]
dc coding
100%|████████████████████████████████| 3201650/3201729 [00:01<00:00, 2014996.9
7it/s]
compressed to 822491 bytes, (or 0.784 MB)
CPU times: user 76.8 ms, sys: 36.3 ms, total: 113 ms
Wall time: 16.5 s
```

In [79]:
```
%%time
!python3 compressor.py result_ov_ar.compressed -c -f decoded_ov_ar.txt
```

```
CPU times: user 157 ms, sys: 35.2 ms, total: 193 ms
Wall time: 29.2 s
```

Проверяем, чтобы файл совпал с исходным:

In [80]:
```
!cmp war_peace_ascii_Zlobin_AS.txt decoded_ov_ar.txt
```

Итого получилось, что исходный текст объёмом 3.05 мб удалось сжать до 1.385 мб кодированием с переполнением, до 0.877 мб дельта кодом, до 0.869 кодом Фибоначчи и до 0.824 мб арифметическим кодированием. Сочетание кодирования с переполнением и арифметического показало наилучший результат в 0.784 мб.

датасет из википедии

In [69]:
```
!python3 compressor.py enwik8.txt -i -f result_enwik8.compressed
```

```
/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%|████████████████████████████████| 99325589/99325589 [00:21<00:00, 4628898.9
0it/s]
dc coding
100%|████████████████████████████████| 99325589/99325687 [00:55<00:00, 1803286.1
6it/s]
compressed to 25596415 bytes, (or 24.411 MB)
```

И сочетанием кодирования с переполнением и арифметического:

In [1]:
```
%%time
!python3 compressor.py enwik8.txt -c -f result_enwik8_ov_ar.compressed
```

```
/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%|████████████████████████████| 99325589/99325589 [00:21<00:00, 4658868.2
5it/s]
dc coding
100%|████████████████████████████| 99325589/99325687 [00:55<00:00, 1802903.1
2it/s]
compressed to 23692592 bytes, (or 22.595 MB)
CPU times: user 2.31 s, sys: 483 ms, total: 2.8 s
Wall time: 7min 52s
```

In [3]:
```
%%time
!python3 compressor.py result_enwik8_ov_ar.compressed -c -f decoded_enwik
```

```
CPU times: user 3.81 s, sys: 890 ms, total: 4.7 s
Wall time: 15min 10s
```

In [4]:
```
!cmp enwik8.txt decoded_enwik8_ov_ar.txt
```

И датасет побольше

Также сочетанием кодирования с переполнением и арифметического:

In [2]:
```
%%time
!python3 compressor.py enwik9.txt -c -f result_enwik9_ov_ar.compressed
```

```
/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%|████████████████████████████| 995619570/995619570 [03:44<00:00, 4431923.7
4it/s]
dc coding
100%|████████████████████████████| 995619570/995619668 [08:54<00:00, 1862359.5
0it/s]
compressed to 188258893 bytes, (or 179.538 MB)
CPU times: user 19 s, sys: 4.36 s, total: 23.3 s
Wall time: 1h 4min 48s
```

Итого, получается, датасет объёмом 95 мб удалось сжать до 22.595 мб, а объёмом 953 мб до 179.538 мб.