## АИСД-2, КДЗ №2

Васильев Владислав Петрович, БПИ 217 В данном .ipynb файле происходит запуск и компиляция кодов с алгоритмами на языке C++, а также рисование сравнительных графиков. Измерение времени происходит в наносекундах

naive - наивный алгоритм

kmp\_standart - алгоритм КПМ с применением стандартных граней kmp\_updated - алгоритм КМП с применением уточненных граней z\_function - алгоритм поиска с помощью Z-функции

Все выводы по графикам находятся в файле "conclusion.pdf"

```
import matplotlib.pyplot as plt
import pandas as pd
import random

plt.rcParams['figure.figsize']=[18,9]
methods = ['naive', 'kmp_standart', 'kmp_updated','z_function']
```

## Генерация строк и шаблонов

Все строки и шаблоны находятся в директории data, в которой содержатся поддиректории:

small\_2 - строка из символов '0', '1' размера 10000 big\_2 - строка из символов '0', '1' размера 100000 small\_4 - строка из символов '0', '1', '2', '3' размера 10000 big\_4 - строка из символов '0', '1', '2', '3' размера 100000

В каждой поддиректории находится файл со строкой, в которой выполняется поиск - search\_string.txt, а также директория templates с шаблонами для данной строки. Шаблоны для і символов подстановок находятся в директории substitution\_{i}, в которой файлы имеют вид template\_{длина шаблона}.txt

## Компиляция программ и заполнение файлов с временными характеристиками

Все директории, соотвествующие алгоритмам поиска, находятся в директории methods

Все файлы с временными характеристиками находятся в тех же директориях, что и .cpp файл с рассматриваемым алгоритмом. Данные файлы расположены в директории time, в каждой из которых есть поддиректории, название которых соответсвует видам рассматриваемых строк. В каждой их вышеуказанных директорий находятся файлы вида substitution\_{i}.txt, данные которых соотвествуют результатам временных измерений работы алгоритмов с шаблонам рассматриваемой длины (от 100 до 3000 с шагом 1000).

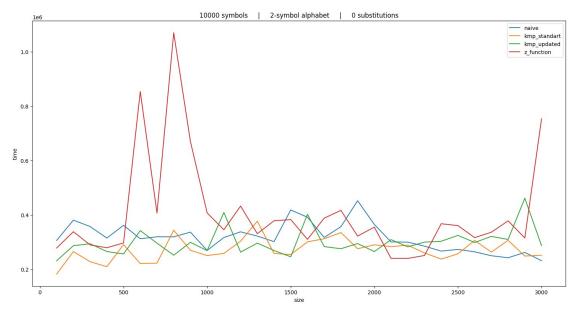
Аргументы компиляции: argv[1] - путь к файлу, текст которого - строка, в которой выполняется поиск argv[2] - путь к файлу, текст которого - шаблон поиска argv[3] - количество подстановок в тексте argv[4] - путь к файлу, в который записывается время работы алгоритма argv[5] - количество запусков работы алгоритма (для усреднения результата по времени) number of repetitions = 100for method in methods: ! g++ -std=c++17 methods/{method}/main.cpp -o methods/{method}/main.out for string in strings: for i in range(5): result path = f"methods/{method}/time/{string}/substitution {i}.txt" search str path = f"data/{string}/search string.txt"

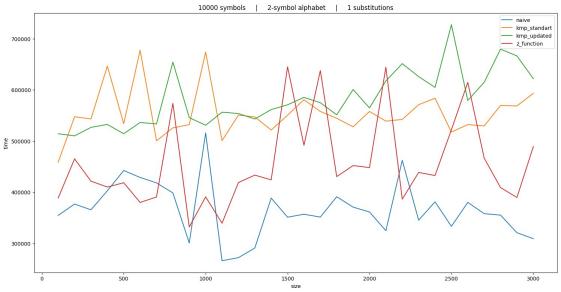
## Сравнение алгоритмов между собой

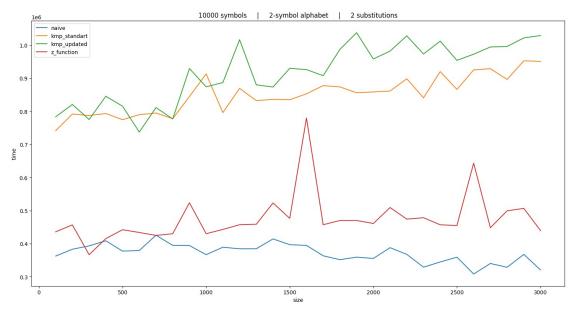
В данной части представлено рисование графиков, на которых происходит сравнение всех алгоритмов по времени работы (в наносекундах) при поиске различных шаблонов в строке.

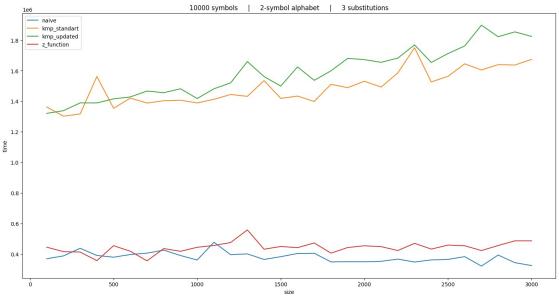
Названия графиков имеют вид "длина строки | длина алфавита | количество подстановок"

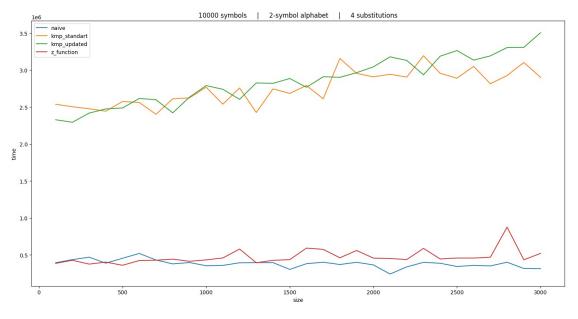
```
for string in strings:
   for i in range(5):
        comparison = \{\}
        comparison["size"] = list(range(100, 3001, 100))
        for method in methods:
open(f"methods/{method}/time/{string}/substitution {i}.txt", 'r') as
file:
                comparison[method] = list(map(int, file.readlines()
[0].split()))
        plot =
pd.DataFrame.from dict(comparison).set index("size").plot()
        plot.set ylabel("time")
        length = "100000" if string[0]=='b' else "10000"
        plot.set title(f"{length} symbols
                                                    {string[-1]}-
                 | {i} substitutions")
symbol alphabet
```

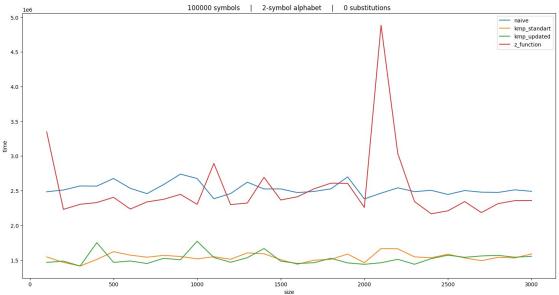


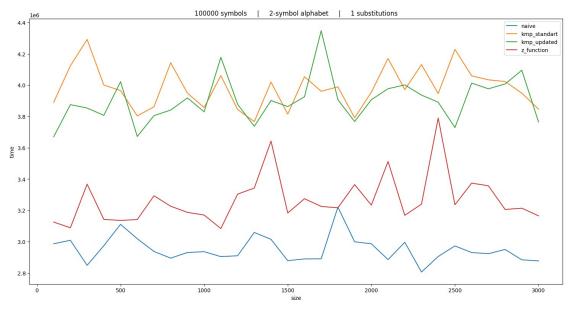


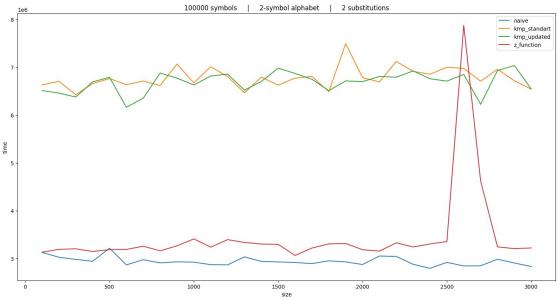


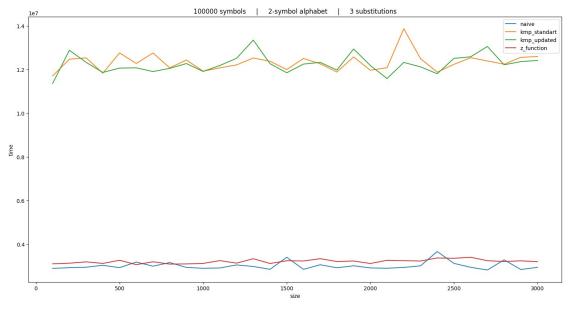


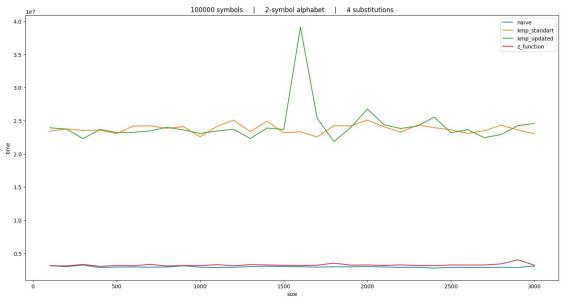


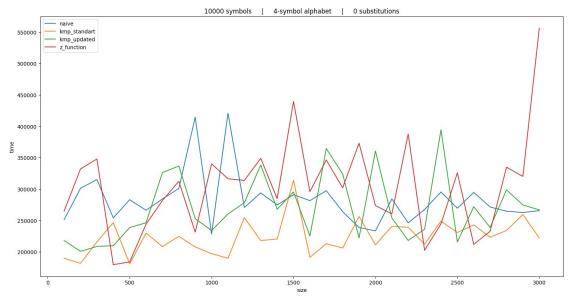


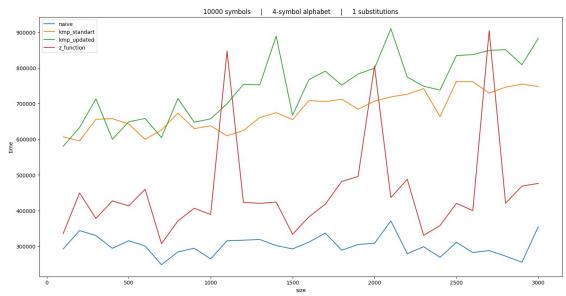


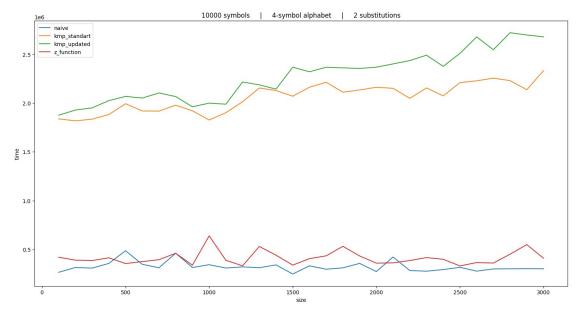


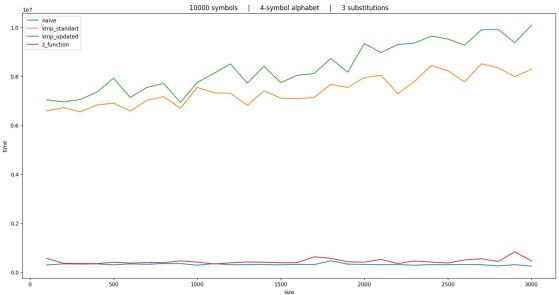


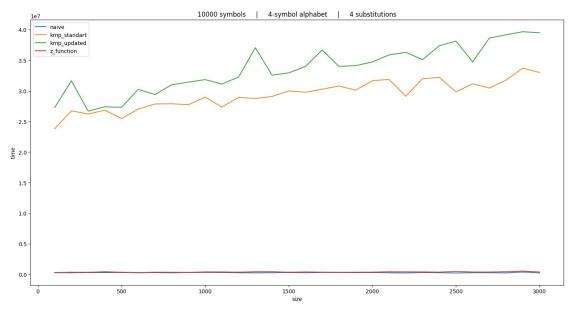


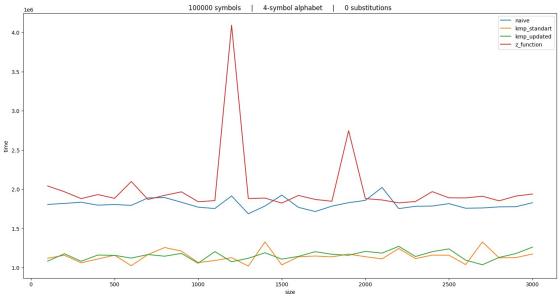


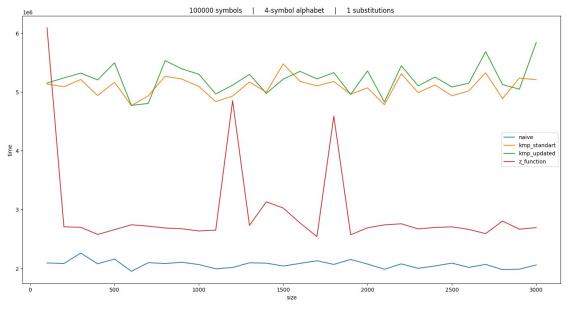


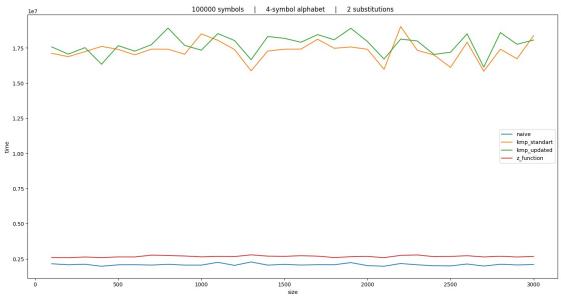


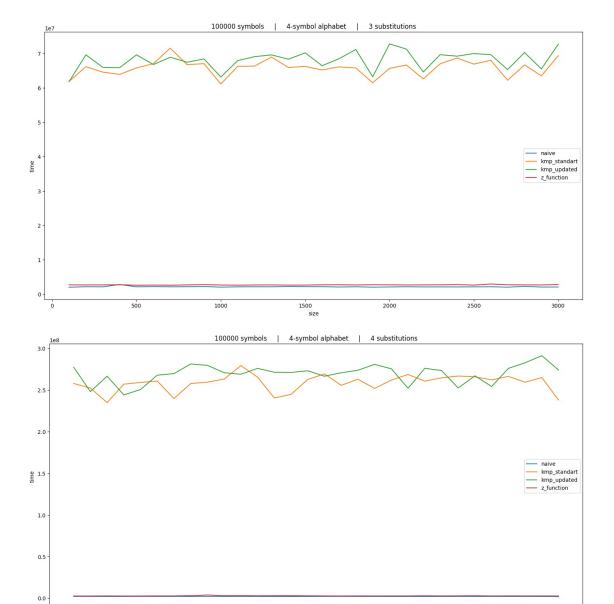












size