Были получены результаты по работе программы, созданы 10 таблиц: по две на каждое изменение шаблонов (по количеству измененных символов на '?') - измеренное время работы и измереннное количество посимвольных сравнений. Шаблоны были сгенерированы для каждого текста, и алгоритмы прогонялись по одинаковым шаблонам. Изменения шаблона происходили так, что заданее генерировались 4 различных индекса строки, на которых будет производиться замена символа на '?'. И постепенно, по одному индексу, шаблоны меняются, и алгоритмы прогоняются по новым шаблонам.

На графиках можно увидеть, что наивный поиск во всех случаях имеет меньшее количество посимвольных сравнений, чем алгоритмы КМП, а также в среднем на него тратится меньше вычислительного времени.

С ростом размера шаблона количество посимвольных сравнений у наивного поиска уменьшается, в то время как у алгоритмов КМП увеличивается. Это связано с тем, что алгоритмам КМП необходимо считать грани и уточненные грани.

На большом бинарном тексте алгоритмы КМП тратят меньше времени на работу, чем наивный поиск.

Между собой стандартный и оптимальный алгоритмы КМП идут примерно вровень, с чуть большим временем выполнения в среднем у оптимального алгоритма.

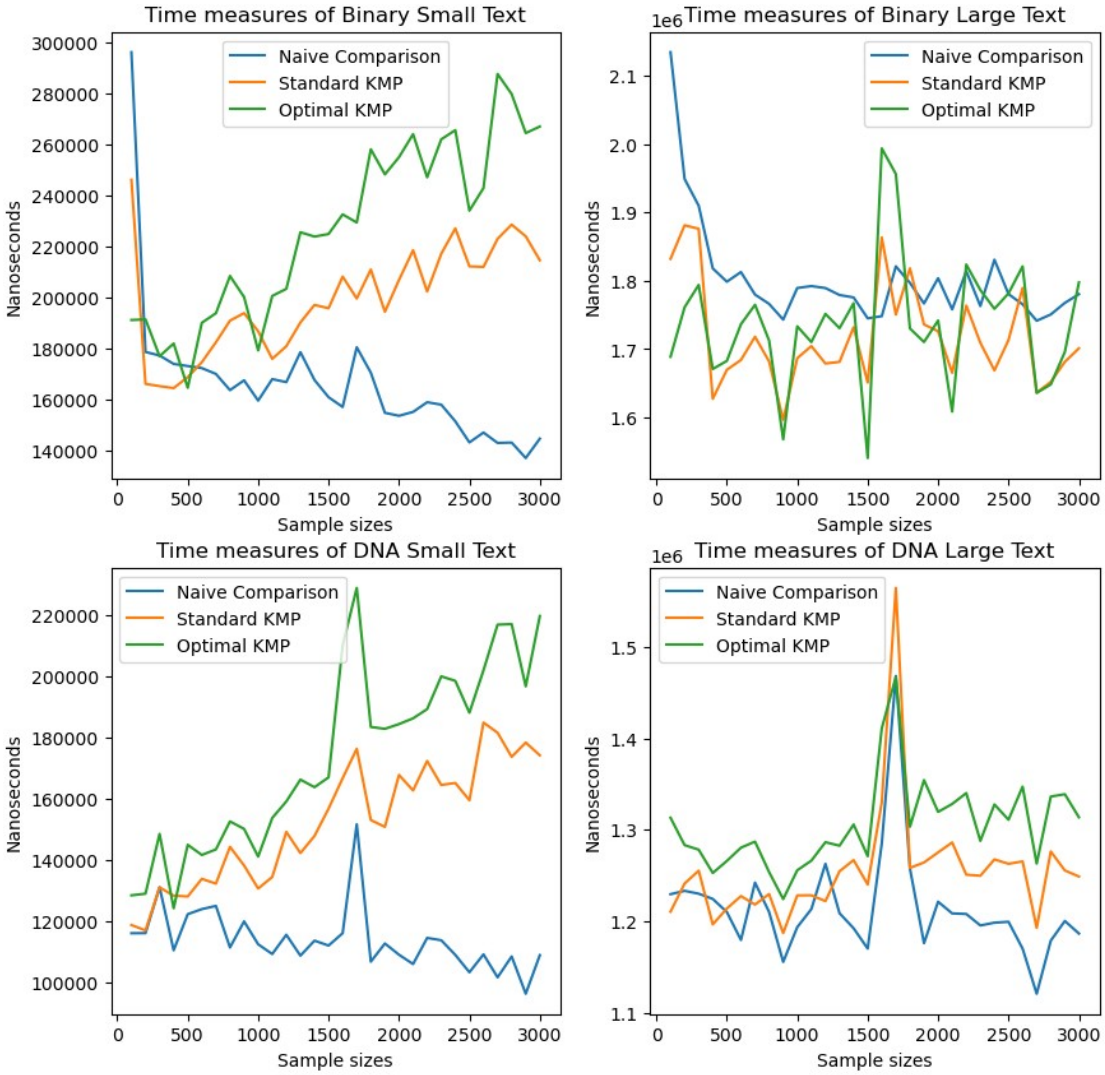
На небольших шаблонах алгоритмы КМП показывают себя лучше в плане времени выполнения (на бинарных текстах), чем наивный поиск, однако с ростом размера шаблона время работы алгоритмов КМП растет, а время наивного поиска уменьшается.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import itertools
import matplotlib.gridspec as gridspec

algs = ["Naive Comparison", "Standard KMP", "Optimal KMP"]
texts = ["Binary Small Text", "Binary Large Text", "DNA Small Text",
"DNA Large Text"]
```

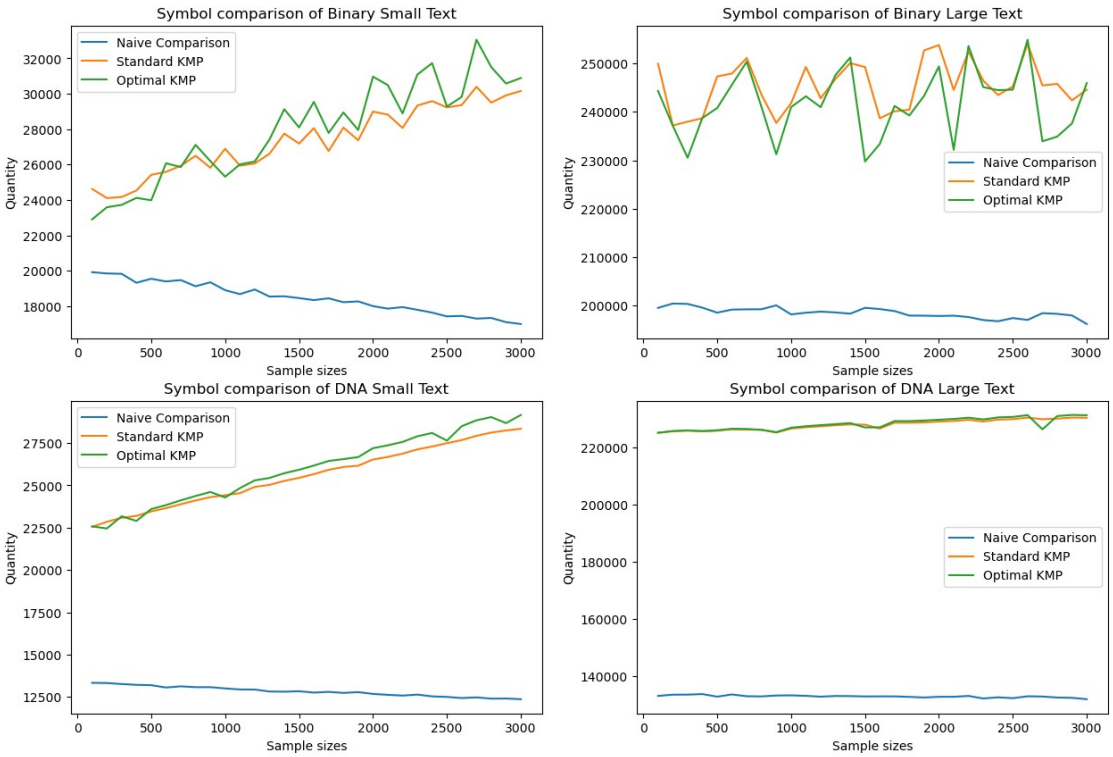
Графики для шаблонов без замены символов на '?'

```
times = pd.read_csv("tables/БПИ213_Кадыкова_0AddsComputingTime.csv",
sep=';')
oper = pd.read_csv("tables/БПИ213_Кадыкова_0AddsOperations.csv",
sep=';')
```

```
#@title Графики измеренного времени по всем алгоритмам для конкретного
текста, где шаблоны оставлены неизменными
#for all algs, computing time table with 0 adds of '?' in samples

gs = gridspec.GridSpec(2, 2)
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(10, 10))
for curr, grd in zip(texts, itertools.product([0, 1], [0, 1])):
    ax = plt.subplot(gs[grd[0], grd[1]])
    for alg in algs:
        plt.plot(times["Sample sizes"], times[curr + " & " + alg],
label=alg)
    ax.set(xlabel='Sample sizes', ylabel='Nanoseconds',
        title='Time measures of ' + curr)
    plt.legend()

fig.savefig("images/ZeroAddsComputingTime.png")
plt.show()
```



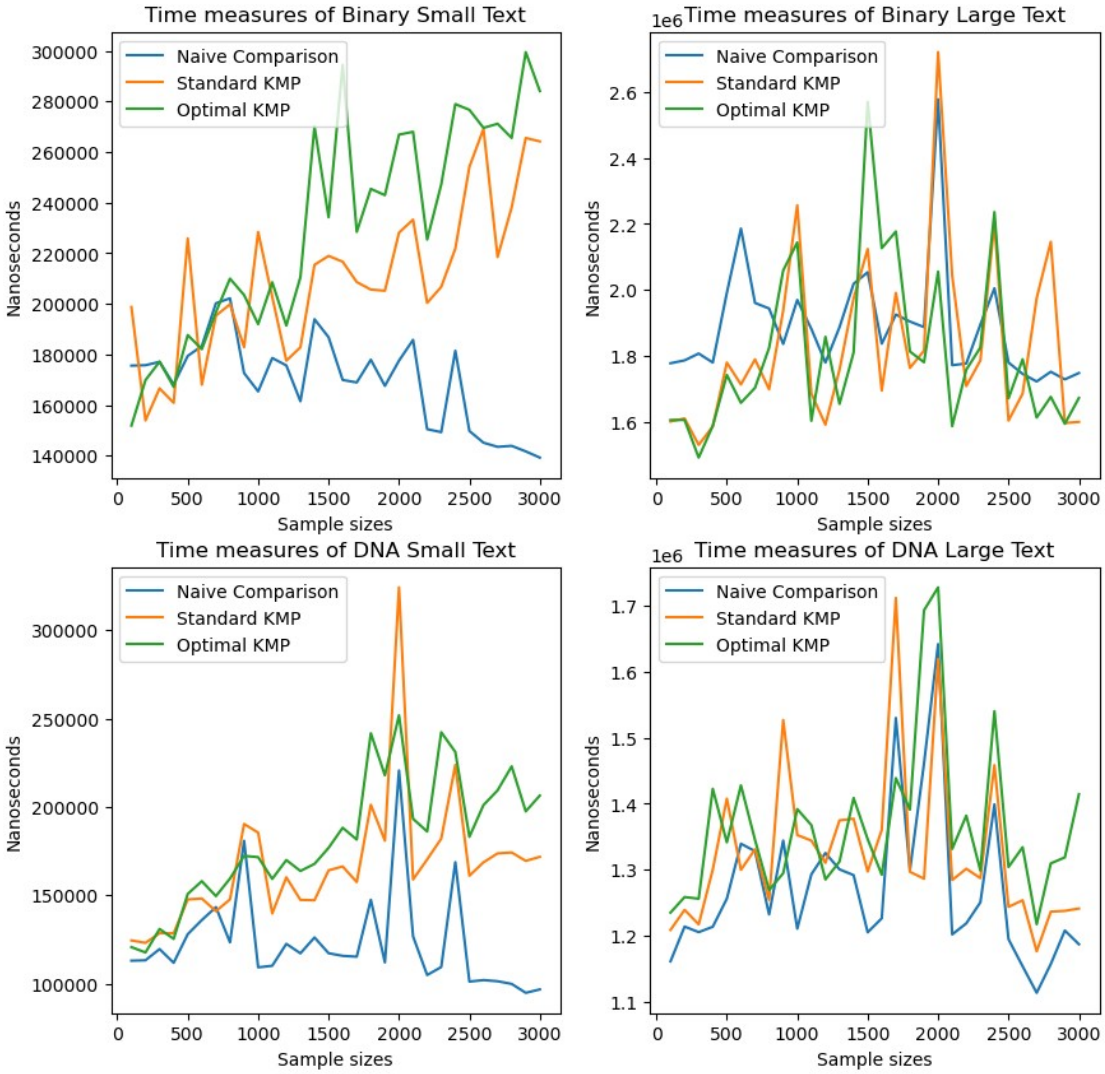
```
#@title Графики количества посимвольных сравнений по всем алгоритмам
для конкретного текста, где шаблоны оставлены неизменными
#for all algs, operations table with 0 adds of '?' in samples

gs = gridspec.GridSpec(2, 2)
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(15, 10))
for curr, grd in zip(texts, itertools.product([0, 1], [0, 1])):
    ax = plt.subplot(gs[grd[0], grd[1]])
    for alg in algs:
        plt.plot(oper["Sample sizes"], oper[curr + " & " + alg],
label=alg)
    ax.set(xlabel='Sample sizes', ylabel='Quantity',
        title='Symbol comparison of ' + curr)
    plt.legend()

fig.savefig("images/ZeroAddsOperations.png")
plt.show()
```



Графики для шаблонов с 1 измененным символом

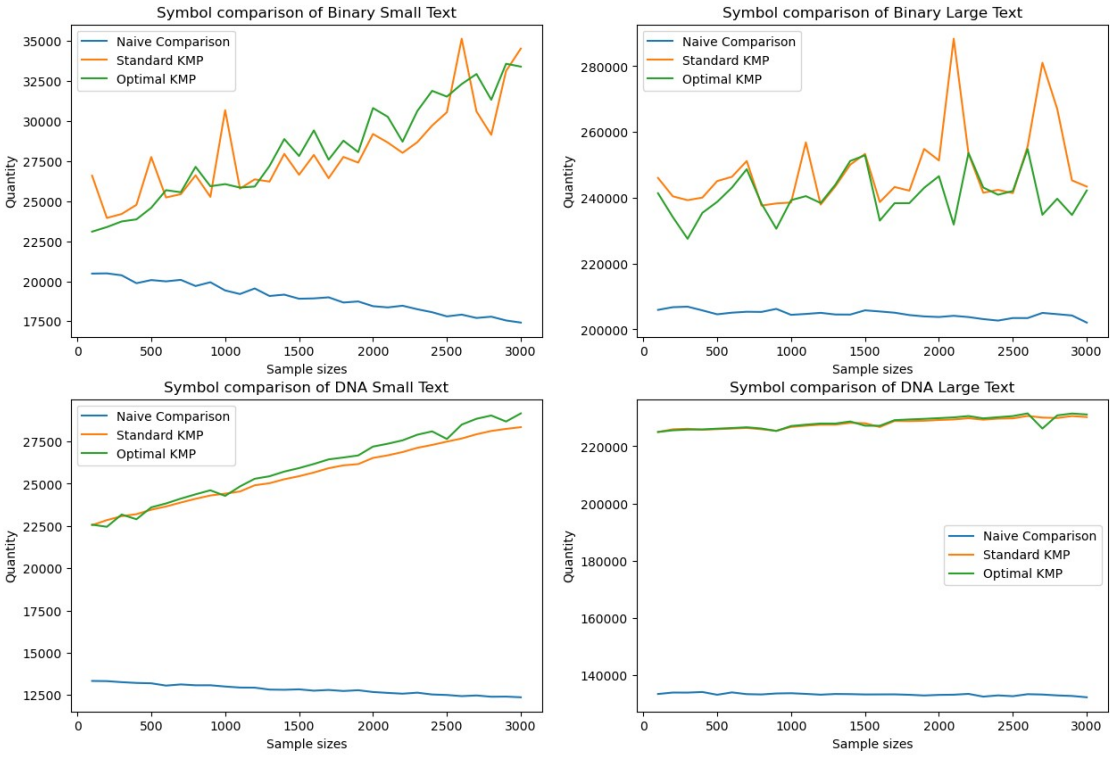
```
times = pd.read_csv("tables/БПИ213_Кадыкова_1AddsComputingTime.csv",
sep=';')
oper = pd.read_csv("tables/БПИ213_Кадыкова_1AddsOperations.csv",
sep=';')

#@title Графики измеренного времени по всем алгоритмам для конкретного
текста, где в шаблоне заменен 1 символ на '?'
#for all algs, computing time table with 1 add of '?' in samples
```

```
gs = gridspec.GridSpec(2, 2)
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(10, 10))
for curr, grd in zip(texts, itertools.product([0, 1], [0, 1])):
    ax = plt.subplot(gs[grd[0], grd[1]])
    for alg in algs:
        plt.plot(times["Sample sizes"], times[curr + " & " + alg],
label=alg)
    ax.set(xlabel='Sample sizes', ylabel='Nanoseconds',
        title='Time measures of ' + curr)
    plt.legend()

fig.savefig("images/OneAddComputingTime.png")
plt.show()
```




```
#@title Графики количества посимвольных сравнений по всем алгоритмам
для конкретного текста, где в шаблоне заменен 1 символ на '?'
#for all algs, operations table with 1 add of '?' in samples
```

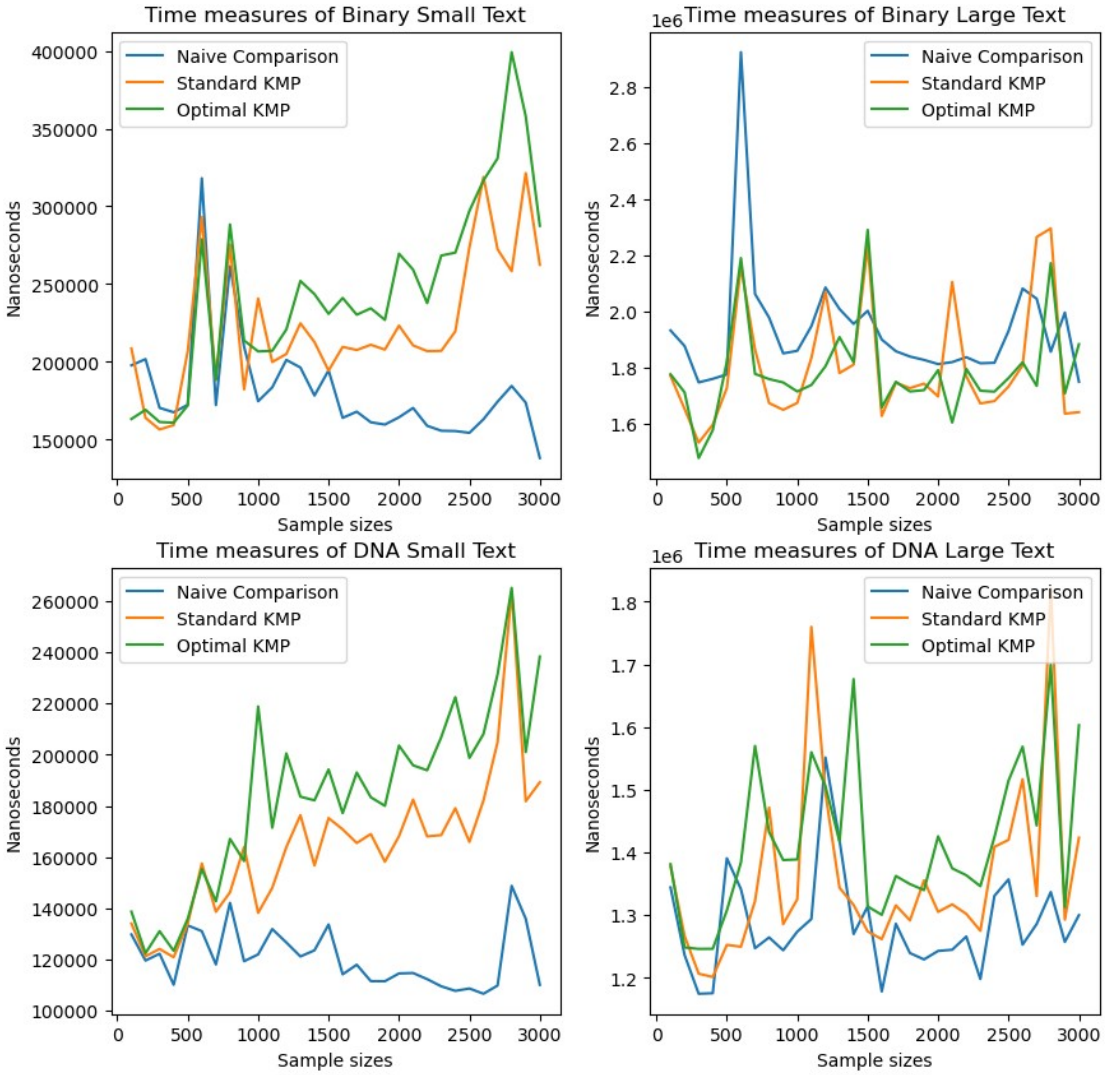
```
gs = gridspec.GridSpec(2, 2)
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(15, 10))
for curr, grd in zip(texts, itertools.product([0, 1], [0, 1])):
    ax = plt.subplot(gs[grd[0], grd[1]])
    for alg in algs:
        plt.plot(oper["Sample sizes"], oper[curr + " & " + alg],
label=alg)
    ax.set(xlabel='Sample sizes', ylabel='Quantity',
        title='Symbol comparison of ' + curr)
    plt.legend()

fig.savefig("images/OneAddOperations.png")
plt.show()
```



Графики для шаблонов с 2 измененными символами

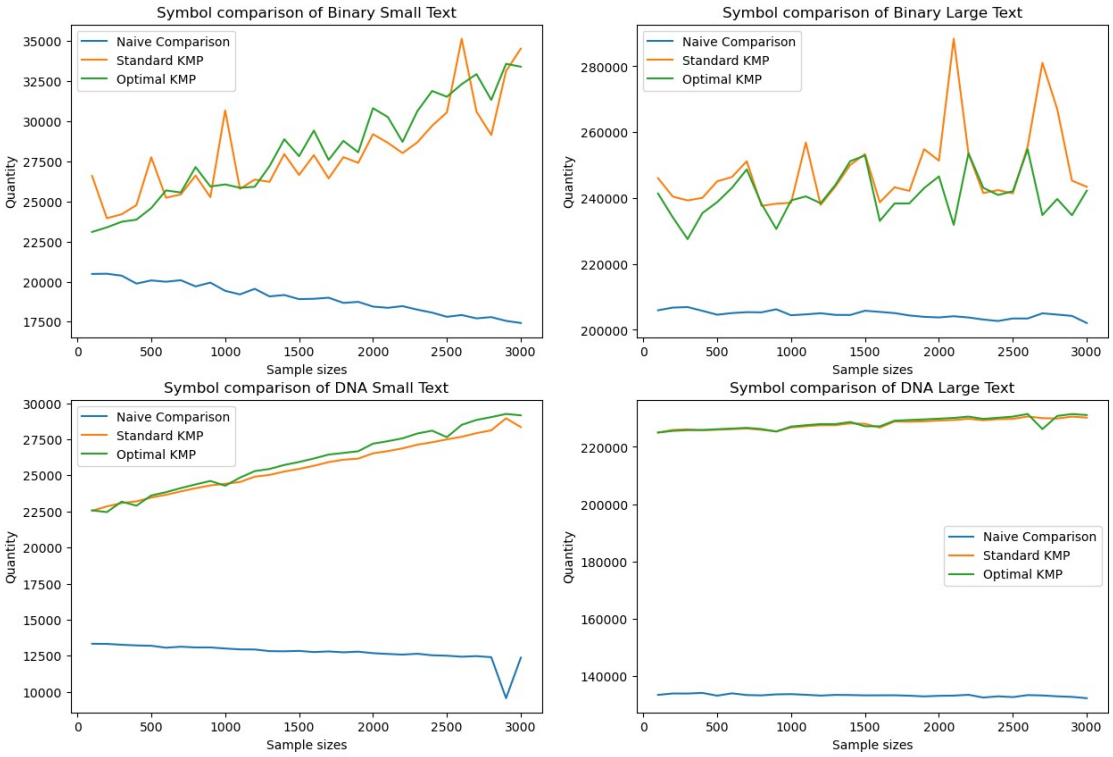
```
times = pd.read_csv("tables/БПИ213_Кадыкова_2AddsComputingTime.csv",
sep=';')
oper = pd.read_csv("tables/БПИ213_Кадыкова_2AddsOperations.csv",
sep=';')

#@title Графики измеренного времени по всем алгоритмам для конкретного
текста, где в шаблоне заменены 2 символа на '?'
#for all algs, computing time table with 2 add of '?' in samples

gs = gridspec.GridSpec(2, 2)
```

```
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(10, 10))
for curr, grd in zip(texts, itertools.product([0, 1], [0, 1])):
    ax = plt.subplot(gs[grd[0], grd[1]])
    for alg in algs:
        plt.plot(times["Sample sizes"], times[curr + " & " + alg],
label=alg)
    ax.set(xlabel='Sample sizes', ylabel='Nanoseconds',
        title='Time measures of ' + curr)
    plt.legend()

fig.savefig("images/TwoAddsComputingTime.png")
plt.show()
```




```
#@title Графики количества посимвольных сравнений по всем алгоритмам
для конкретного текста, где в шаблоне заменены 2 символа на '?'
#for all algs, operations table with 2 adds of '?' in samples

gs = gridspec.GridSpec(2, 2)
```

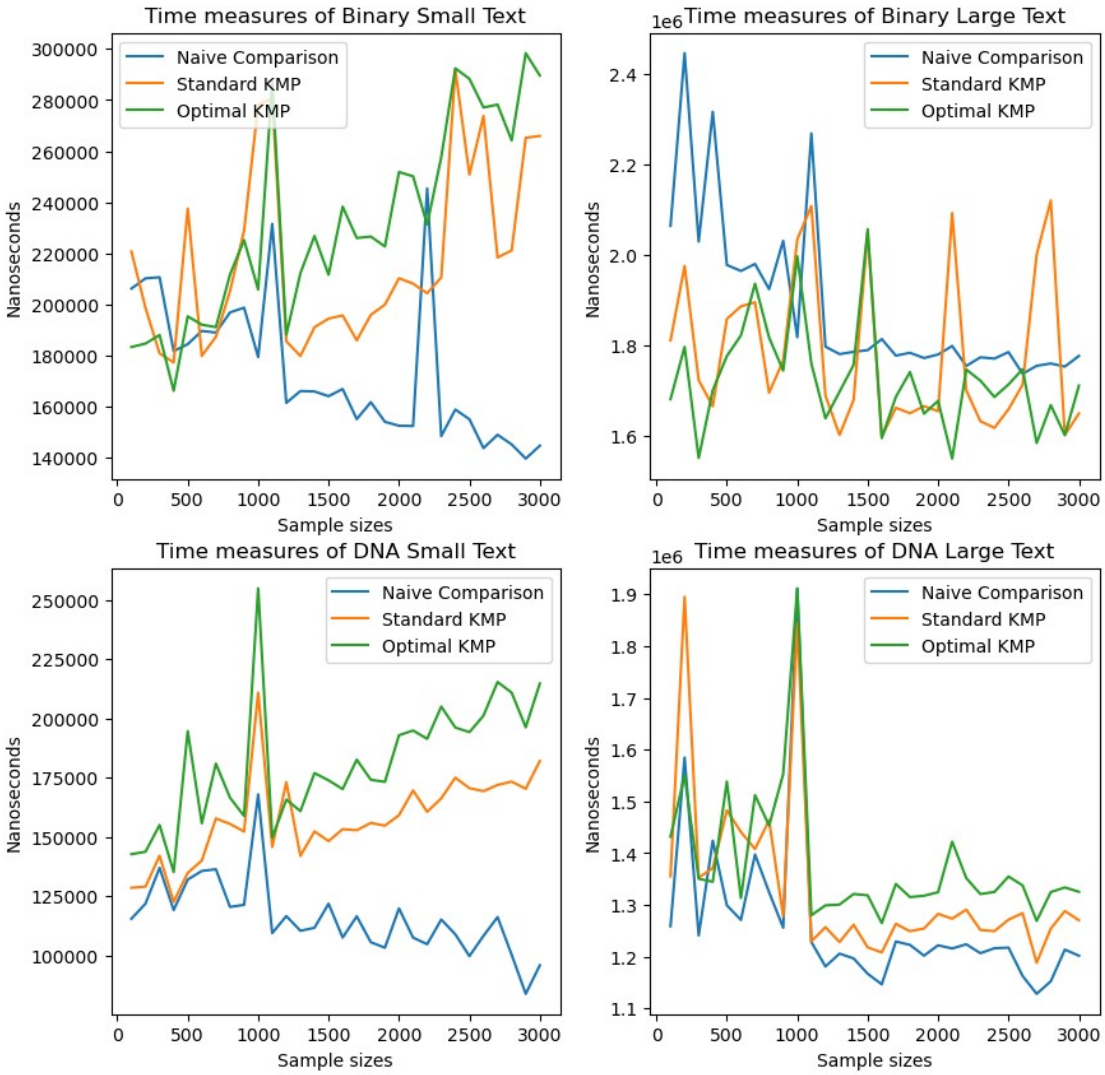
```
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(15, 10))
for curr, grd in zip(texts, itertools.product([0, 1], [0, 1])):
    ax = plt.subplot(gs[grd[0], grd[1]])
    for alg in algs:
        plt.plot(oper["Sample sizes"], oper[curr + " & " + alg],
label=alg)
    ax.set(xlabel='Sample sizes', ylabel='Quantity',
        title='Symbol comparison of ' + curr)
    plt.legend()

fig.savefig("images/TwoAddsOperations.png")
plt.show()
```



Графики для шаблонов с 3 измененным символом

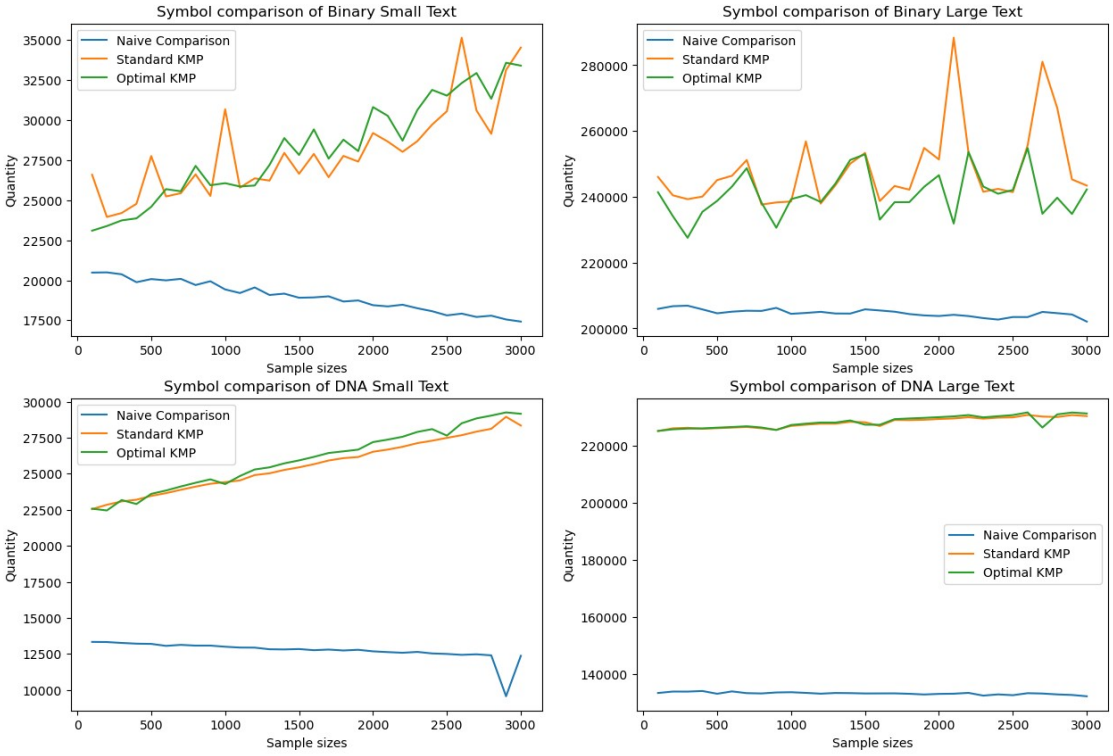
```
times = pd.read_csv("tables/БПИ213_Кадыкова_3AddsComputingTime.csv",
sep=';')
oper = pd.read_csv("tables/БПИ213_Кадыкова_3AddsOperations.csv",
sep=';')

#@title Графики измеренного времени по всем алгоритмам для конкретного
текста, где в шаблоне заменены 3 символа на '?'
#for all algs, computing time table with 3 add of '?' in samples

gs = gridspec.GridSpec(2, 2)
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(10, 10))
for curr, grd in zip(texts, itertools.product([0, 1], [0, 1])):
    ax = plt.subplot(gs[grd[0], grd[1]])
```

```
    for alg in algs:
        plt.plot(times["Sample sizes"], times[curr + " & " + alg],
label=alg)
    ax.set(xlabel='Sample sizes', ylabel='Nanoseconds',
        title='Time measures of ' + curr)
    plt.legend()

fig.savefig("images/ThreeAddsComputingTime.png")
plt.show()
```




```
#@title Графики количества посимвольных сравнений по всем алгоритмам
для конкретного текста, где в шаблоне заменены 3 символа на '?'
#for all algs, operations table with 3 adds of '?' in samples

gs = gridspec.GridSpec(2, 2)
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(15, 10))
for curr, grd in zip(texts, itertools.product([0, 1], [0, 1])):
    ax = plt.subplot(gs[grd[0], grd[1]])
```

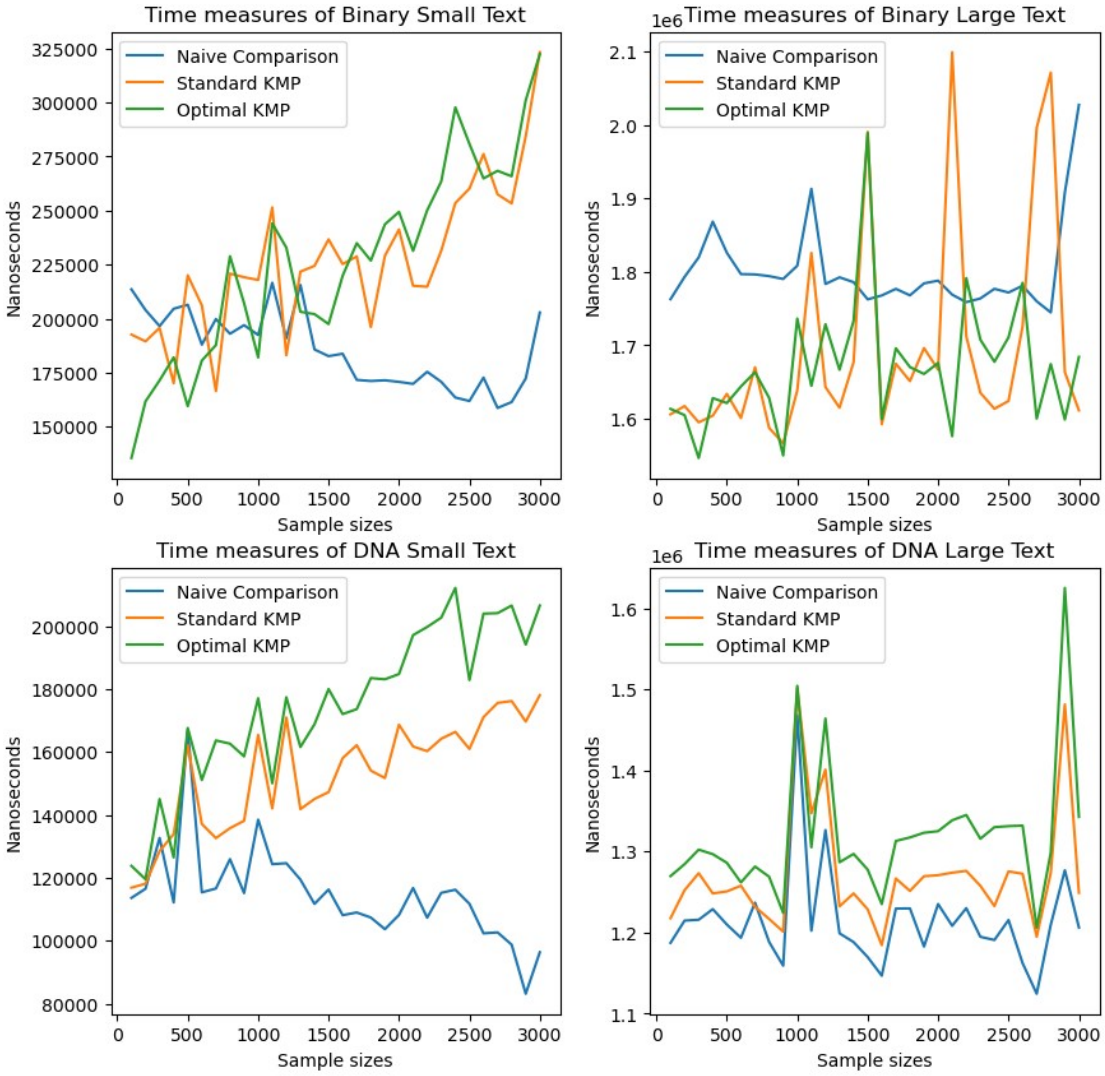
```
    for alg in algs:
        plt.plot(oper["Sample sizes"], oper[curr + " & " + alg],
label=alg)
    ax.set(xlabel='Sample sizes', ylabel='Quantity',
        title='Symbol comparison of ' + curr)
    plt.legend()

fig.savefig("images/ThreeAddsOperations.png")
plt.show()
```



Графики для шаблонов с 4 измененным символом

```
times = pd.read_csv("tables/БПИ213_Кадыкова_4AddsComputingTime.csv",
sep=';')
oper = pd.read_csv("tables/БПИ213_Кадыкова_4AddsOperations.csv",
sep=';')

#@title Графики измеренного времени по всем алгоритмам для конкретного
текста, где в шаблоне заменены 4 символа на '?'
#for all algs, computing time table with 4 add of '?' in samples

gs = gridspec.GridSpec(2, 2)
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(10, 10))
for curr, grd in zip(texts, itertools.product([0, 1], [0, 1])):
    ax = plt.subplot(gs[grd[0], grd[1]])
    for alg in algs:
        plt.plot(times["Sample sizes"], times[curr + " & " + alg],
label=alg)
```

```
    ax.set(xlabel='Sample sizes', ylabel='Nanoseconds',
        title='Time measures of ' + curr)
    plt.legend()

fig.savefig("images/FourAddsComputingTime.png")
plt.show()
```




```
#@title Графики количества посимвольных сравнений по всем алгоритмам
для конкретного текста, где в шаблоне заменены 4 символа на '?'
#for all algs, operations table with 4 adds of '?' in samples

gs = gridspec.GridSpec(2, 2)
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(15, 10))
for curr, grd in zip(texts, itertools.product([0, 1], [0, 1])):
    ax = plt.subplot(gs[grd[0], grd[1]])
    for alg in algs:
        plt.plot(oper["Sample sizes"], oper[curr + " & " + alg],
label=alg)
```

```
    ax.set(xlabel='Sample sizes', ylabel='Quantity',
        title='Symbol comparison of ' + curr)
    plt.legend()

fig.savefig("images/FourAddsOperations.png")
plt.show()
```