

# Анализ полученных результатов

АиСД-2, 2023

Хорасанджян Левон Арменович, БПИ218

Код написан и запущен в CLion

Выполнены все задачи:

- 1) Написаны 4 (1 дополнительная) реализации поисков шаблонов в текстах;
- 2) Засечено время работы сортировок
- 3) Посчитано кол-во элементарных операций (сделал дополнительно из интереса);
- 4) Результаты записаны в .csv-файлы, после чего на их основе были сформированы графики с помощью .ipynb-файлов.

Легенда:

Simple – примитивный алгоритм;



Standard – КМП с обычными гранями;

Modified – КМП с уточненными гранями;

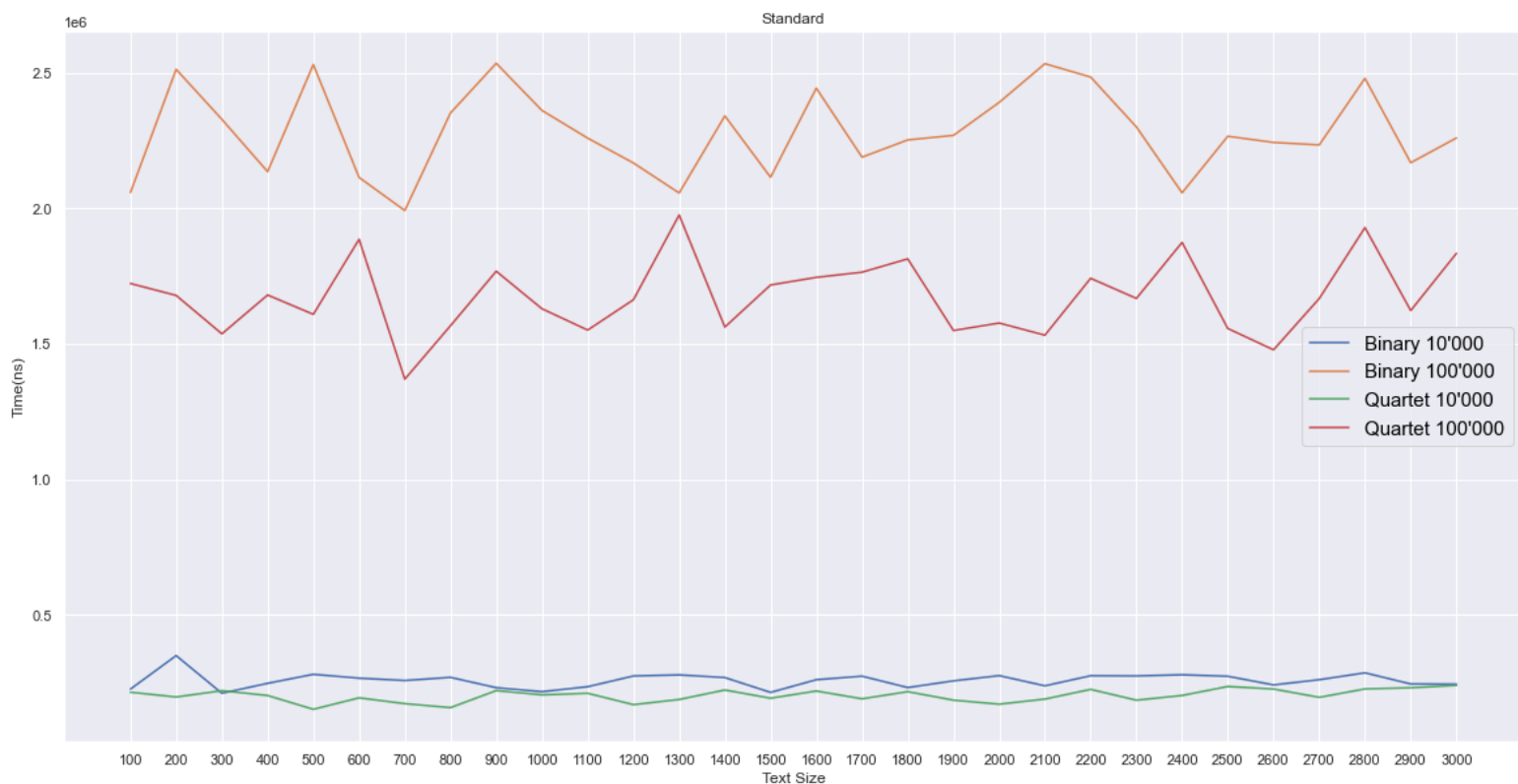
Z-func – Z-функция.

## Подсчёт времени

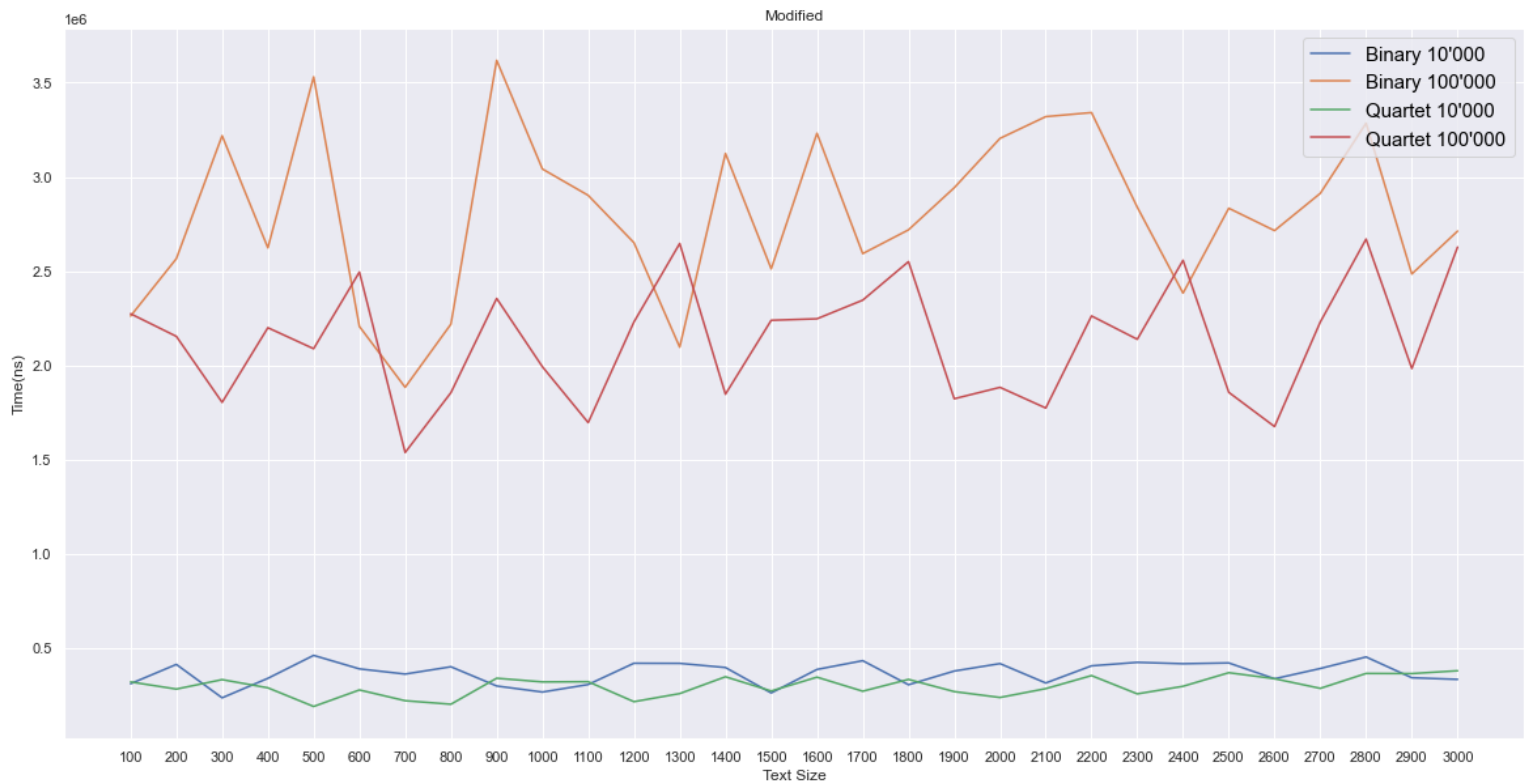
---

*0 знаков подстановки*

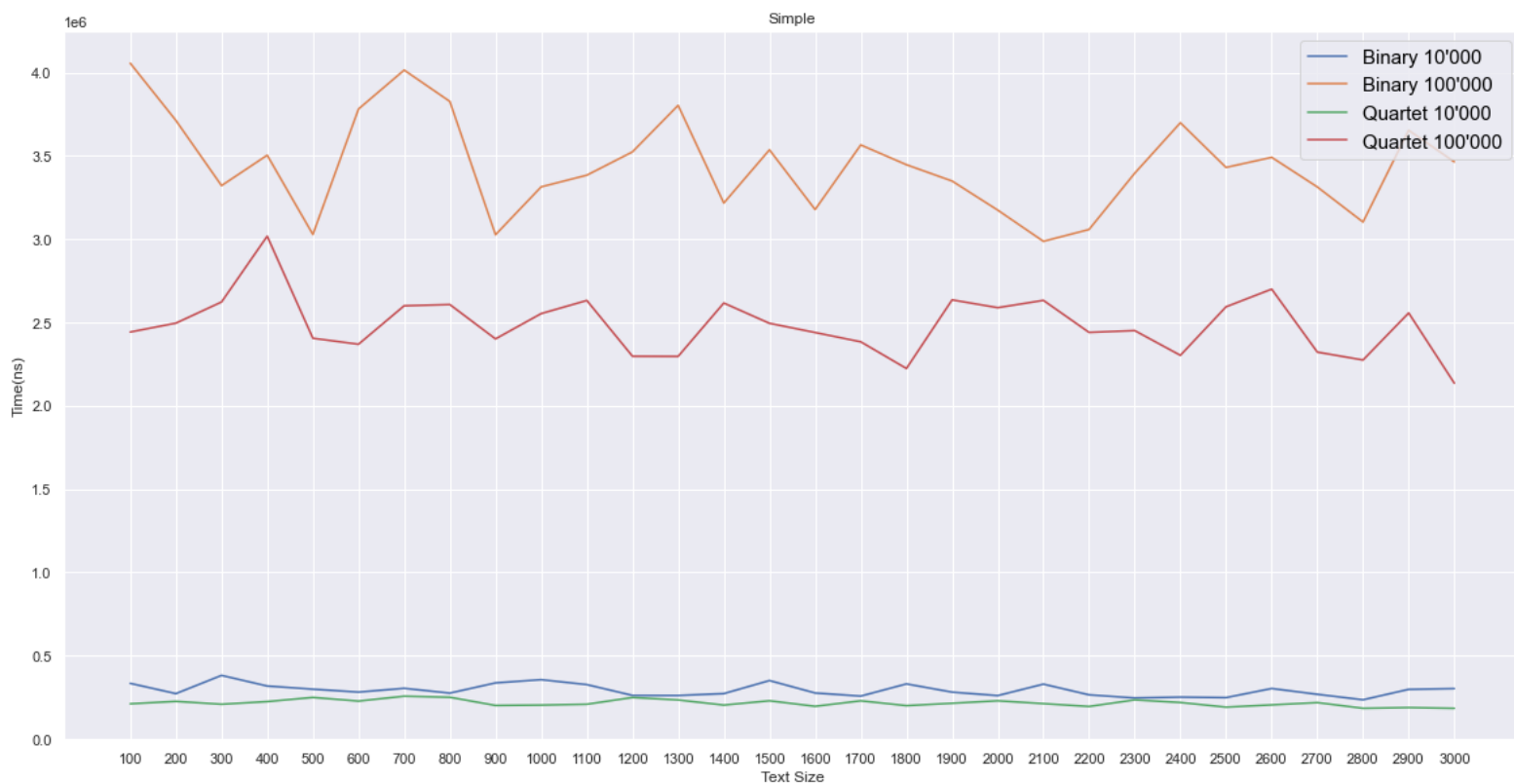
---



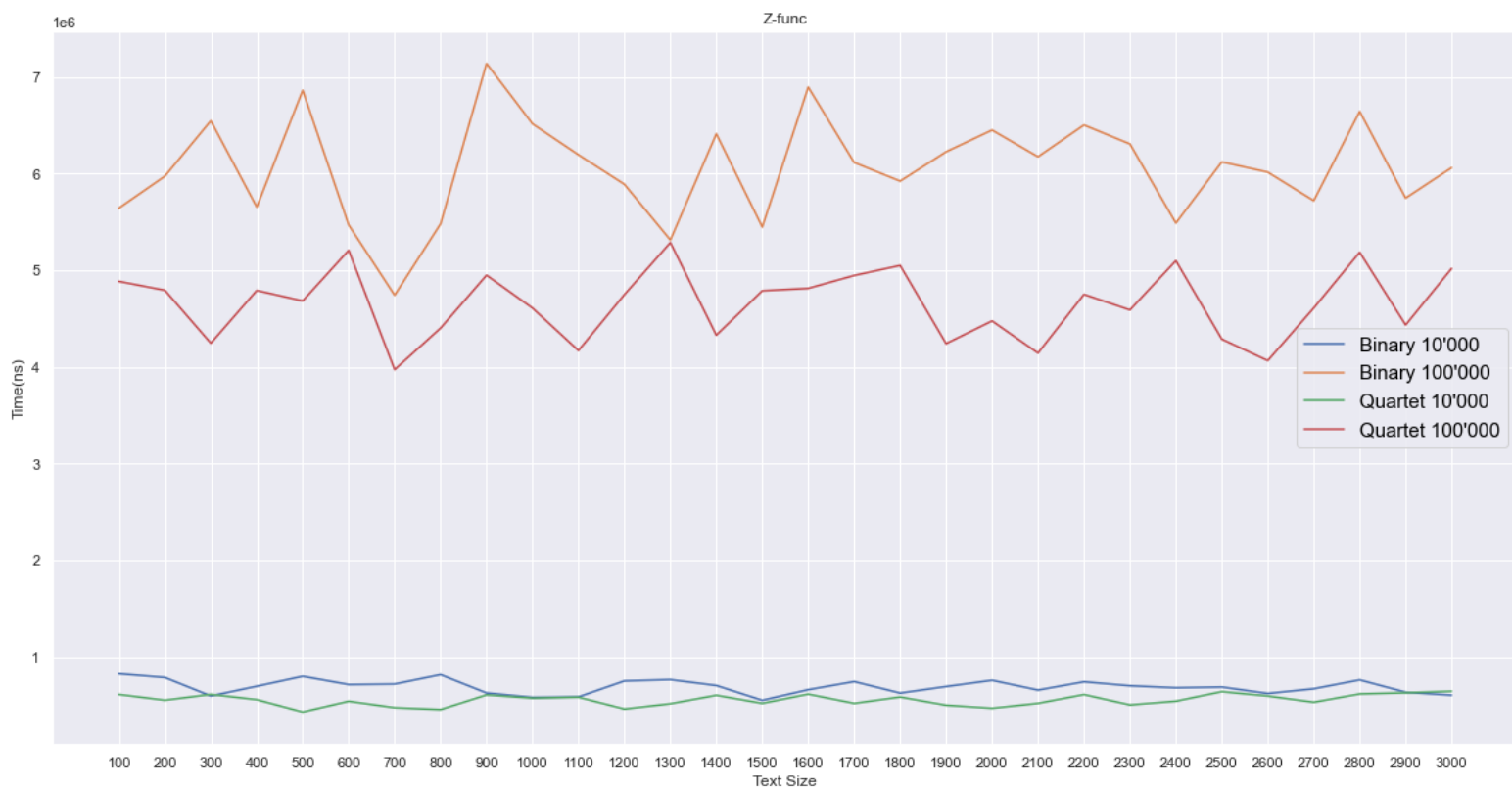
Как и ожидалось, обработка обычного КМП на 100.000 символов с большой разницей занимает больше времени, чем обработка на 10.000 символов. Чем больше размер, тем скачки графиков сильнее, что тоже весьма логично.



При оптимизированном КМП мы наблюдаем ту же ситуацию, однако скачки ещё больше, и местами 100к\_бинарник работает медленнее, чем 100к\_квартет.



Ситуация схожа с обычным КМП, однако скачок между размерами текста 500 и 600 равен около  $0.8 + 1e6$  нс, в то время как у КМП на тех же размерах скачок поменьше и примерно равен  $0.3 + 1e6$  нс.

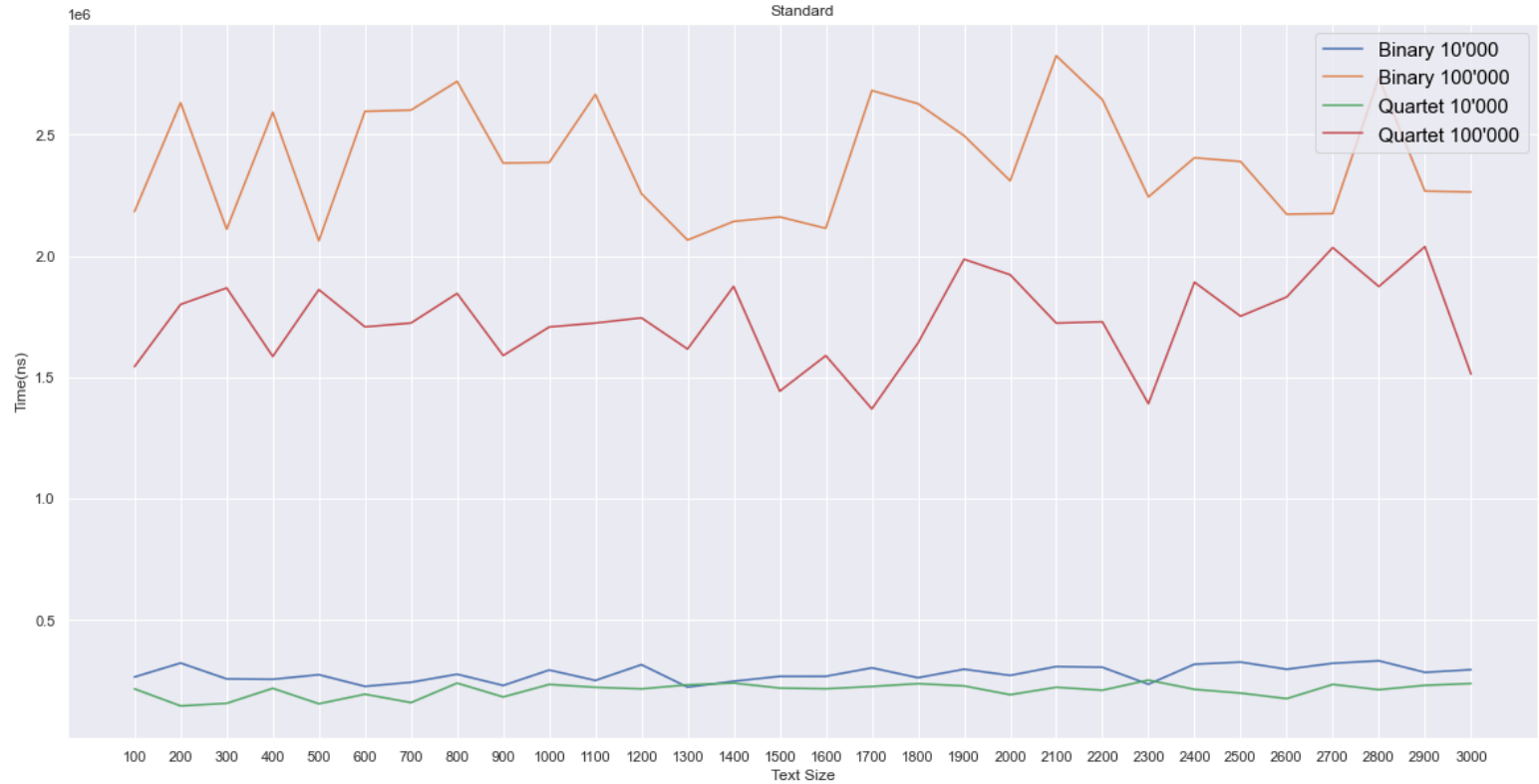


Для Z-функции, как и в случае с другими алгоритмами, снова наблюдаются большие скачки при увеличении размера исходного текста. Также по времени наблюдается самая большая из всех 4-ёх алгоритмов зависимость от размеров исходного текста: на 10к примерно  $0.5 + 1e6$  нс, а на 100к минимальное время  $4 + 1e6$  нс, то есть разница примерно равна  $3.5 + 1e6$  (у других же алгоритмов разница от минимального примерно  $1 + 1e6$  нс или  $2 + 1e6$  нс).

---

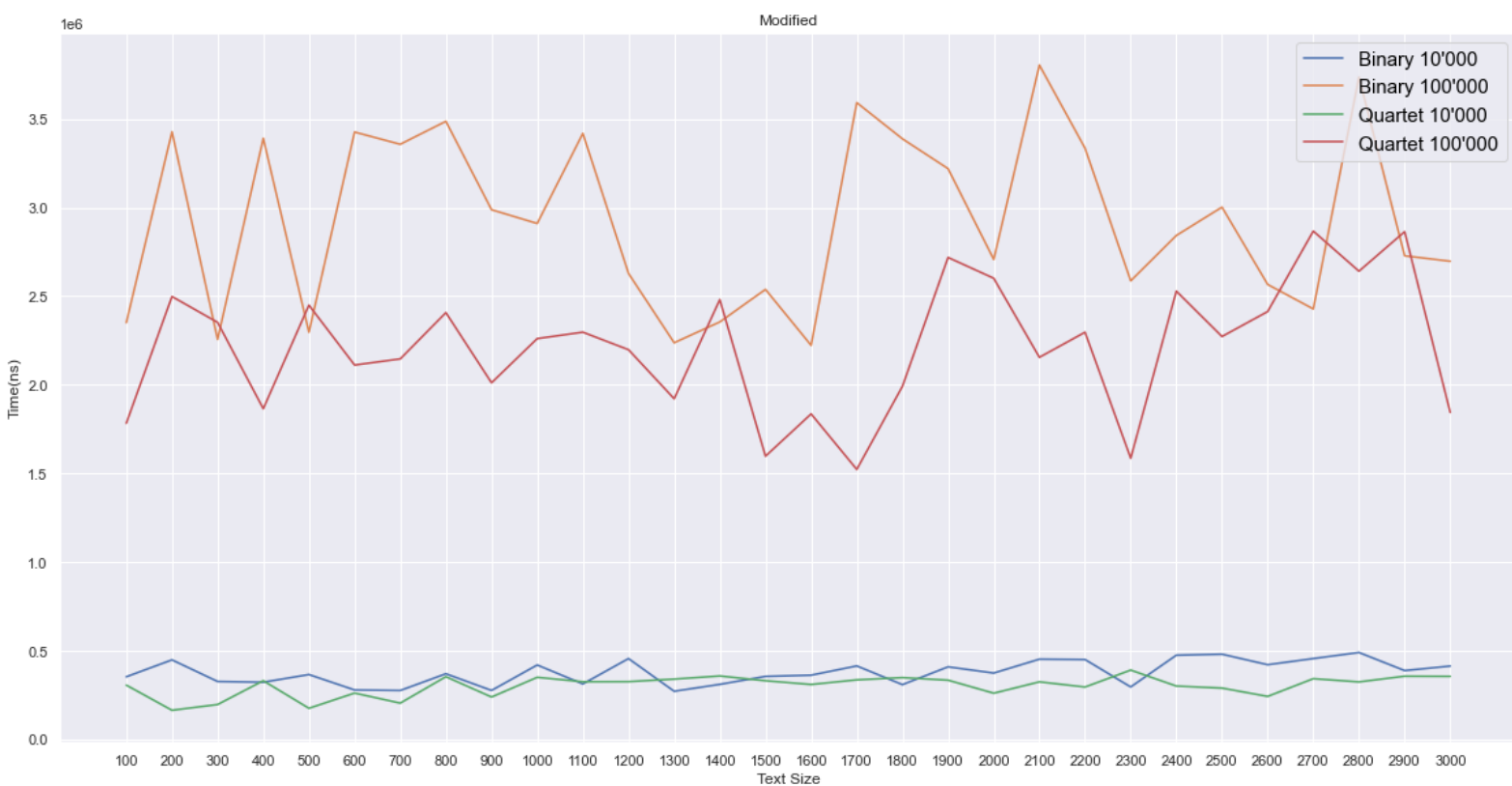
## 1 знак подстановки

---

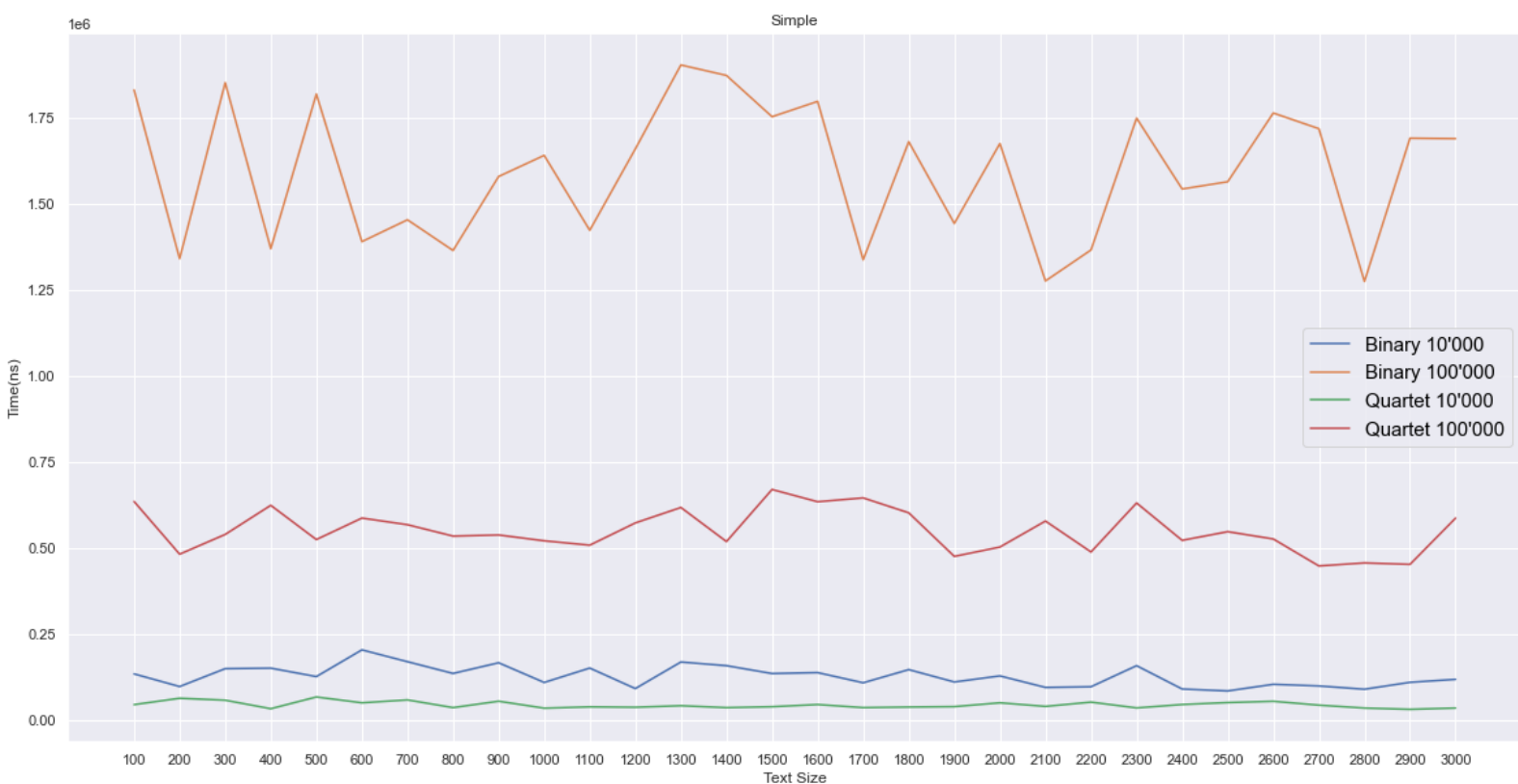


Точно также, как и в аналогичной ситуации, но без символов подстановки, однако на большем кол-ве размеров значение начало превышать за  $2.5 \times 10^6$ .

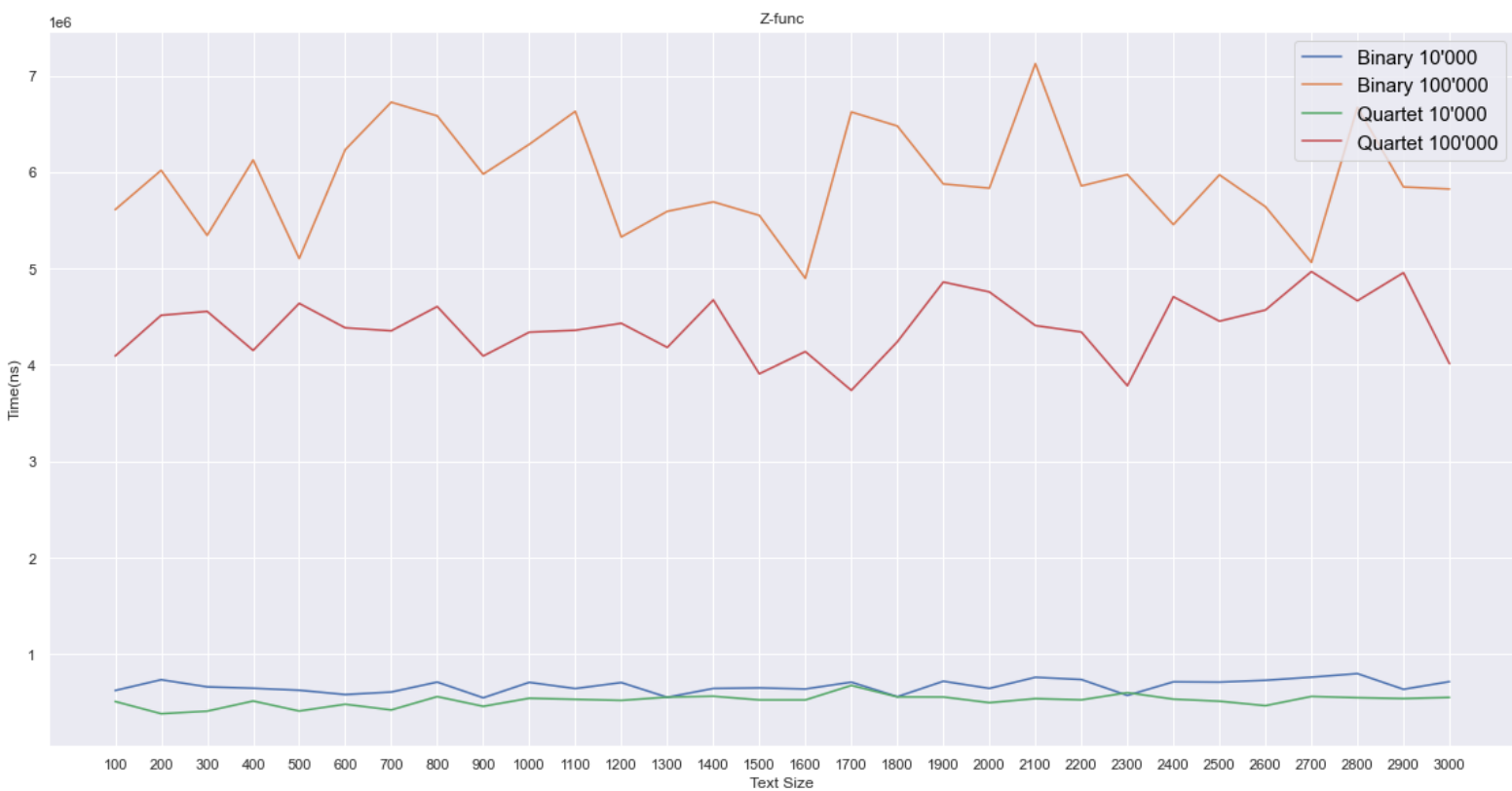




Точно также, как и в аналогичной ситуации, но без символов подстановки, однако на большем кол-ве размеров значение начало превышать за  $3 \cdot 10^6$ .



Все графики спустились ниже и начали показывать время лучше, чем без знака подстановки.

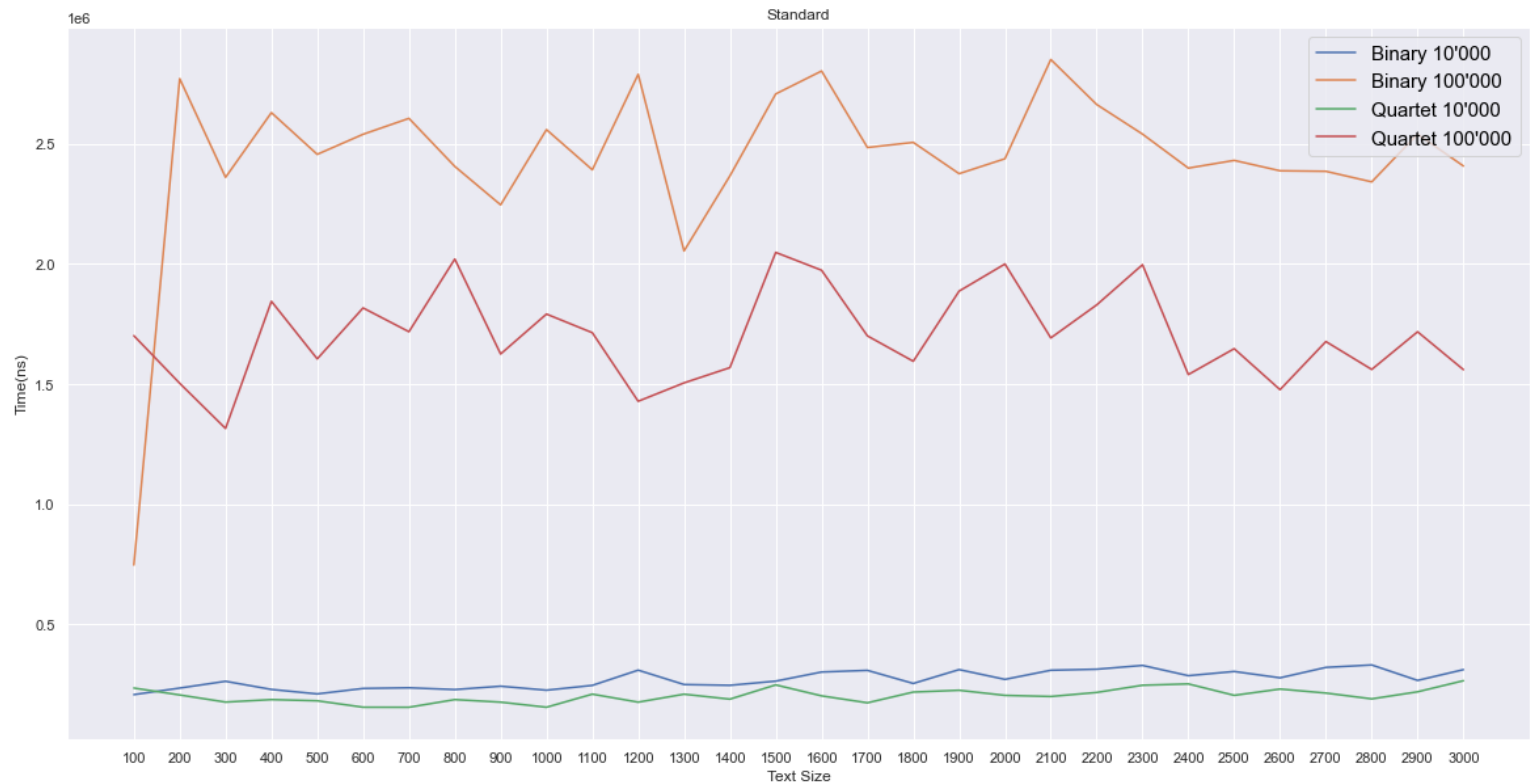


В целом, схожие результаты, как и без символа подстановки.

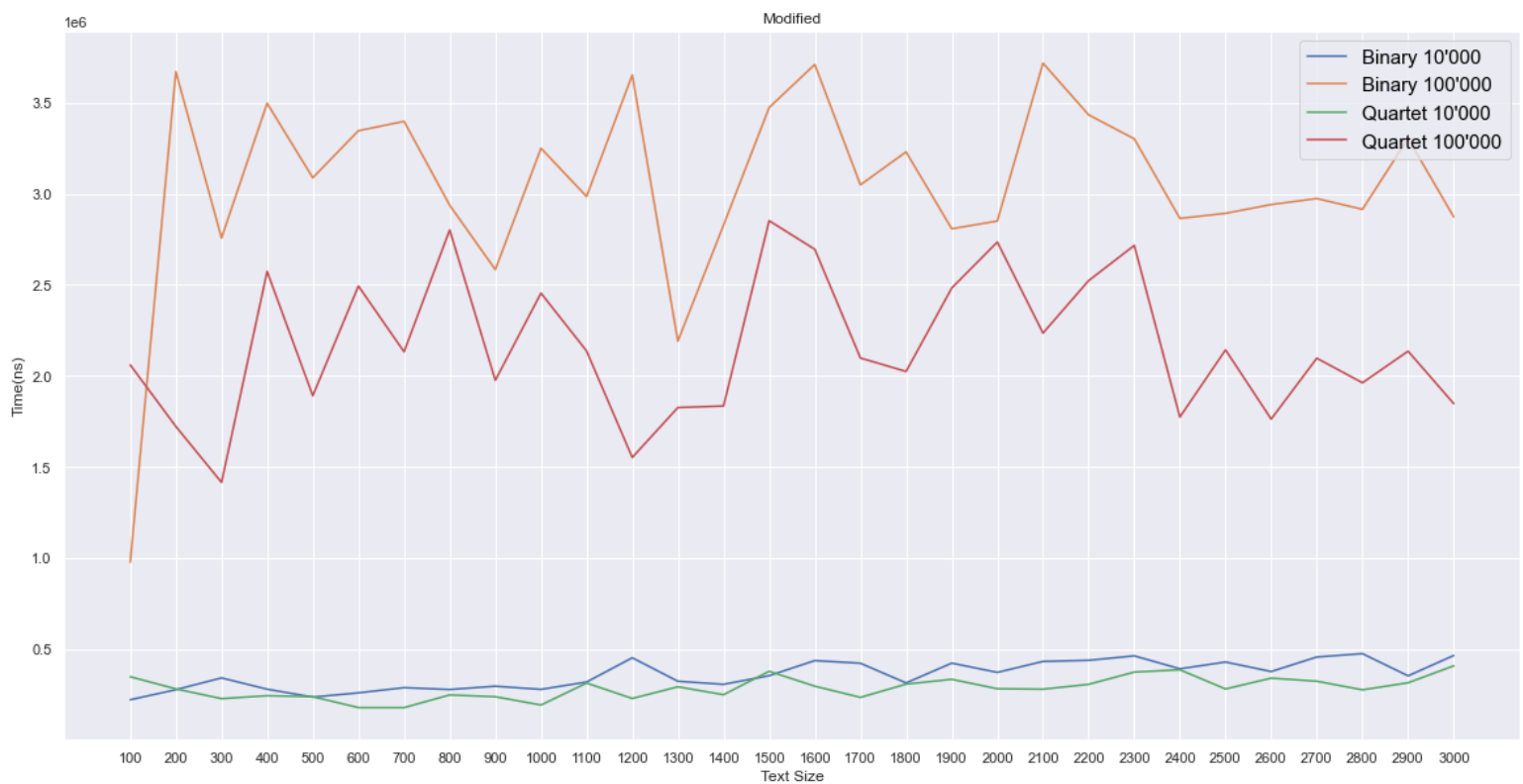
---

## 2 знака подстановки

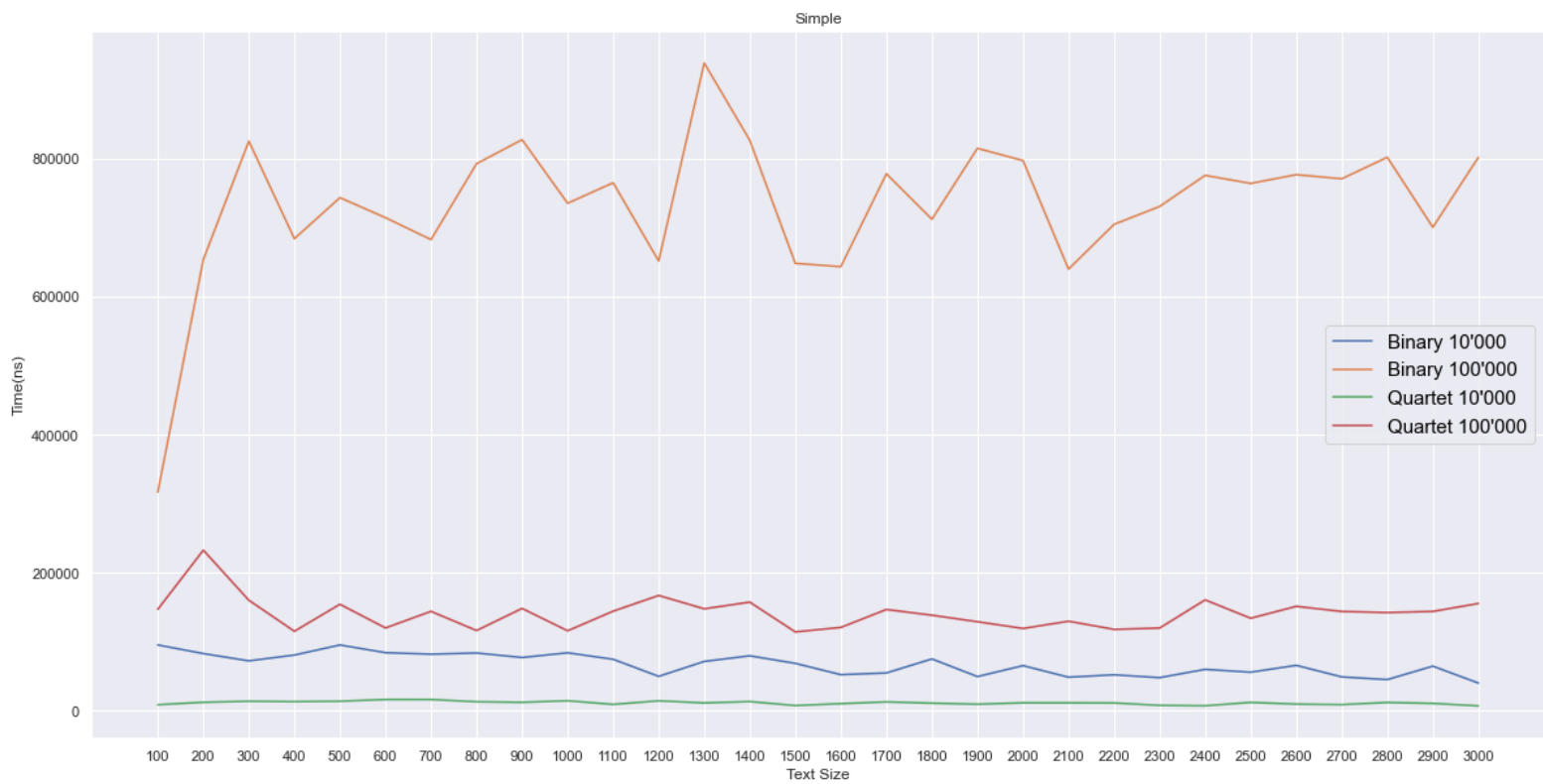
---



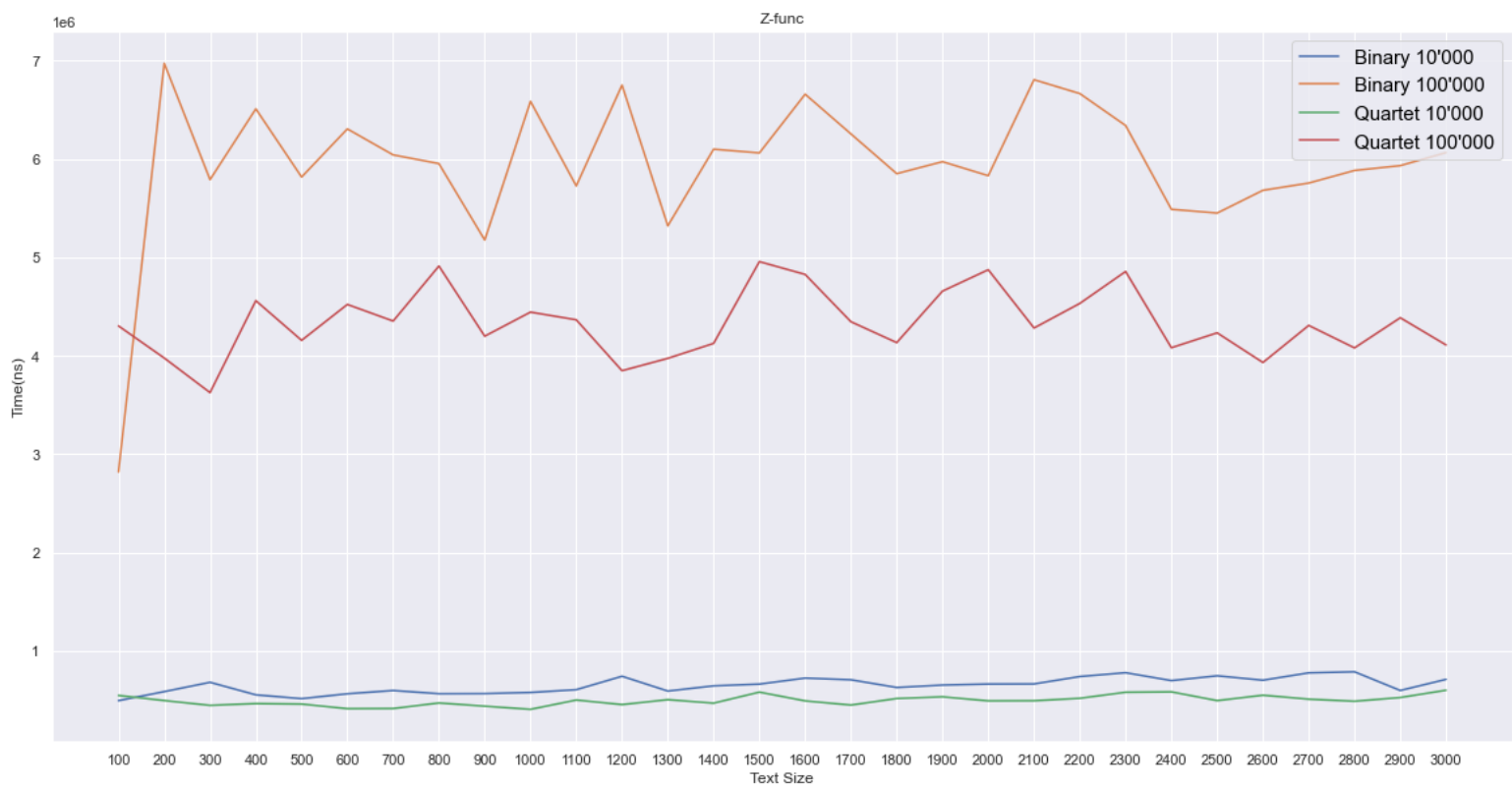
Точно также, как и в аналогичной ситуации, но без символов подстановки, однако на большем кол-ве размеров значение начало превышать за  $2.5 \times 10^6$ .



Точно также, как и в аналогичной ситуации, но без символов подстановки, однако на большем кол-ве размеров значение начало превышать за  $3 \times 10^6$ .

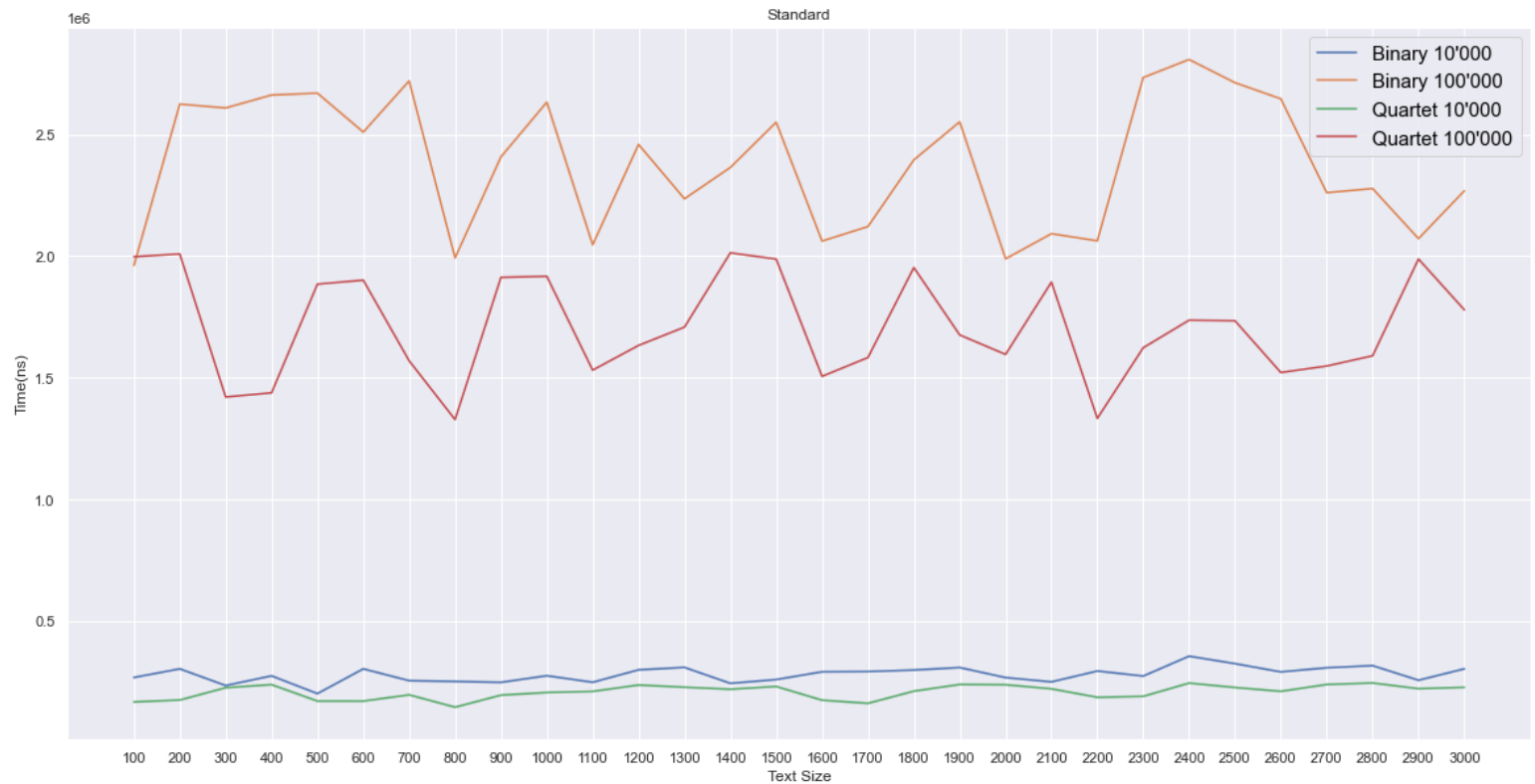


Все графики спустились ниже и начали показывать время лучше, чем с 1 знаком подстановки.



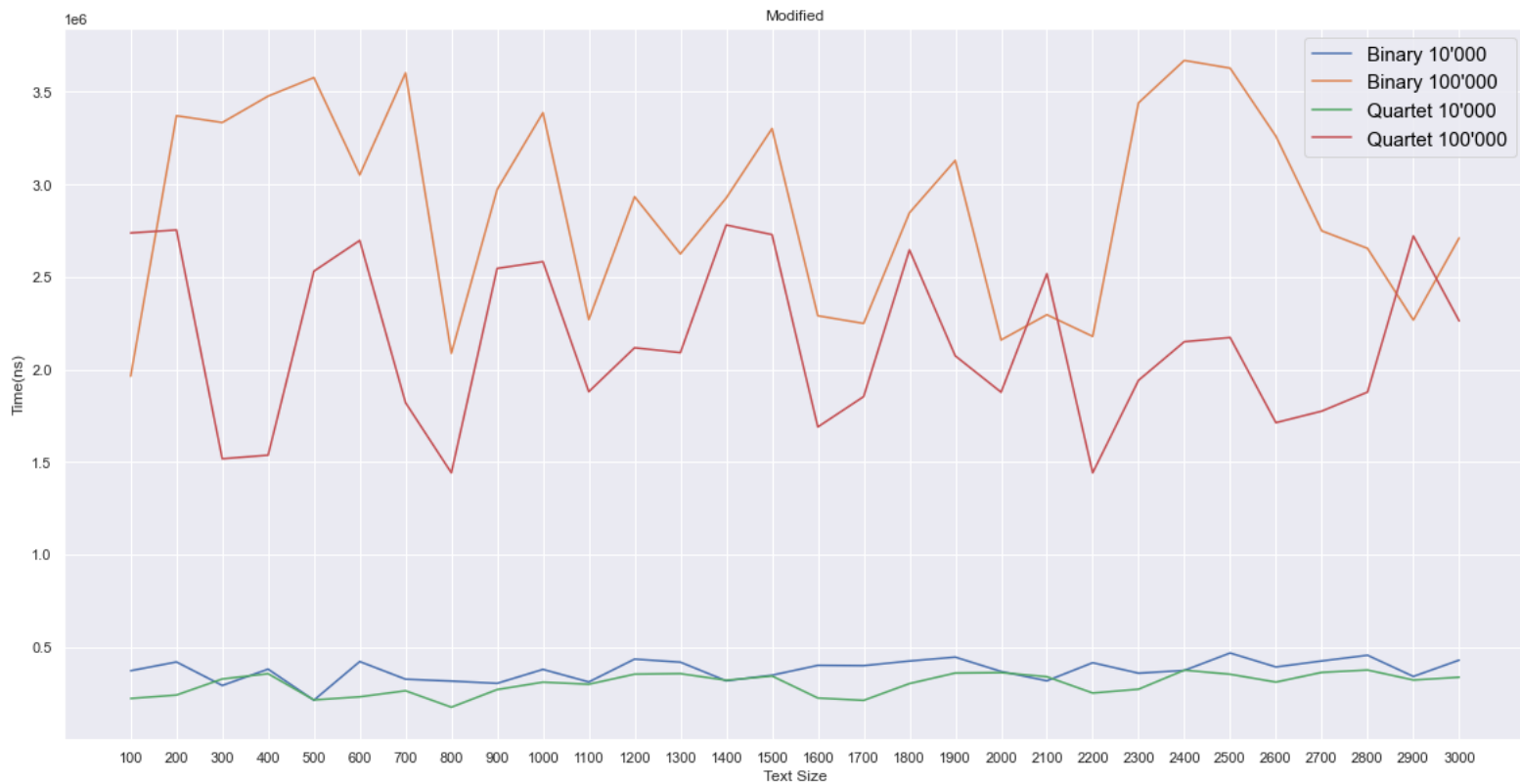
В целом, схожие результаты, как и без символа подстановки.

### 3 знака подстановки

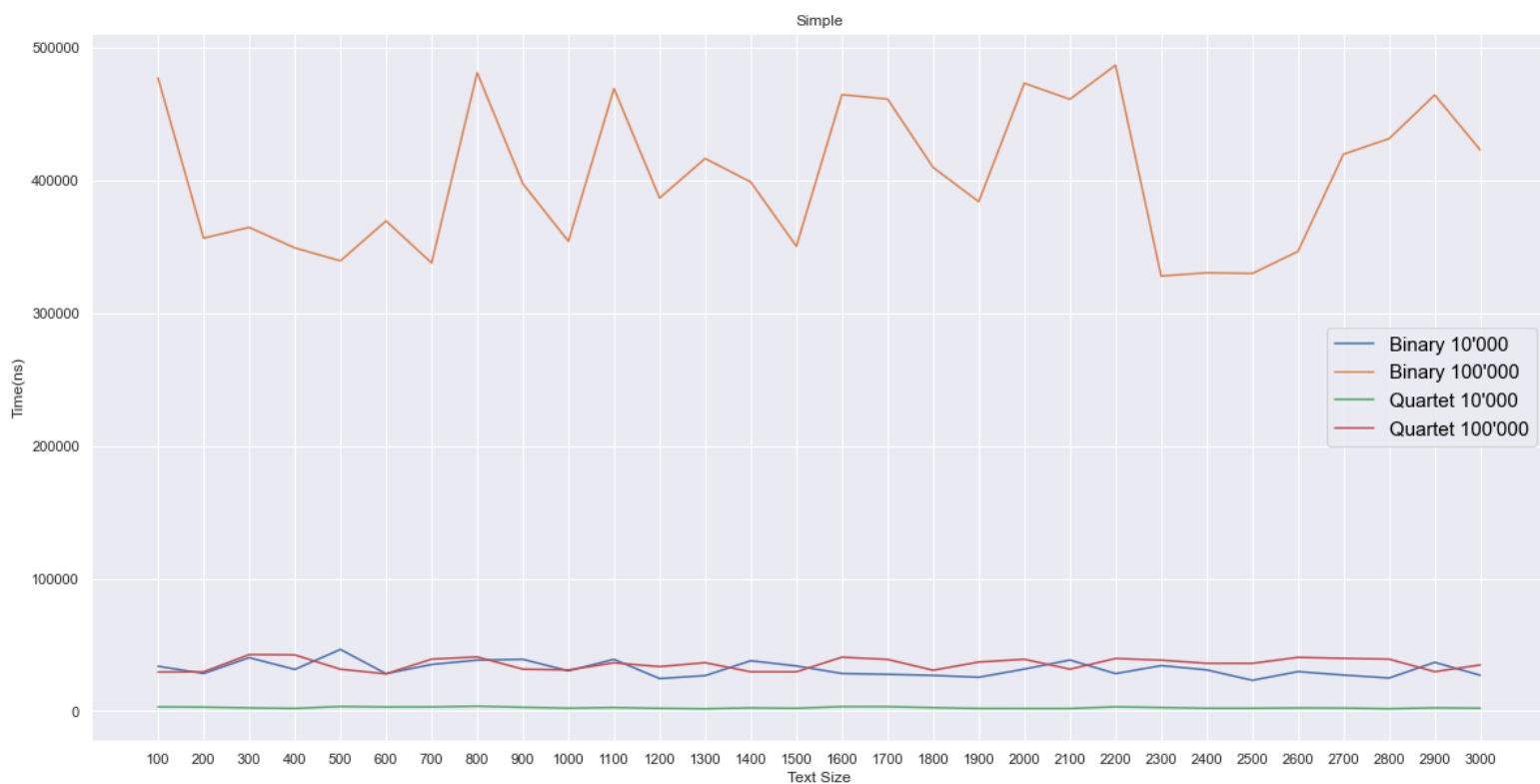


Точно также, как и в аналогичной ситуации, но без символов подстановки, однако на большем кол-ве размеров значение начало превышать за  $2.5 \cdot 10^6$ .

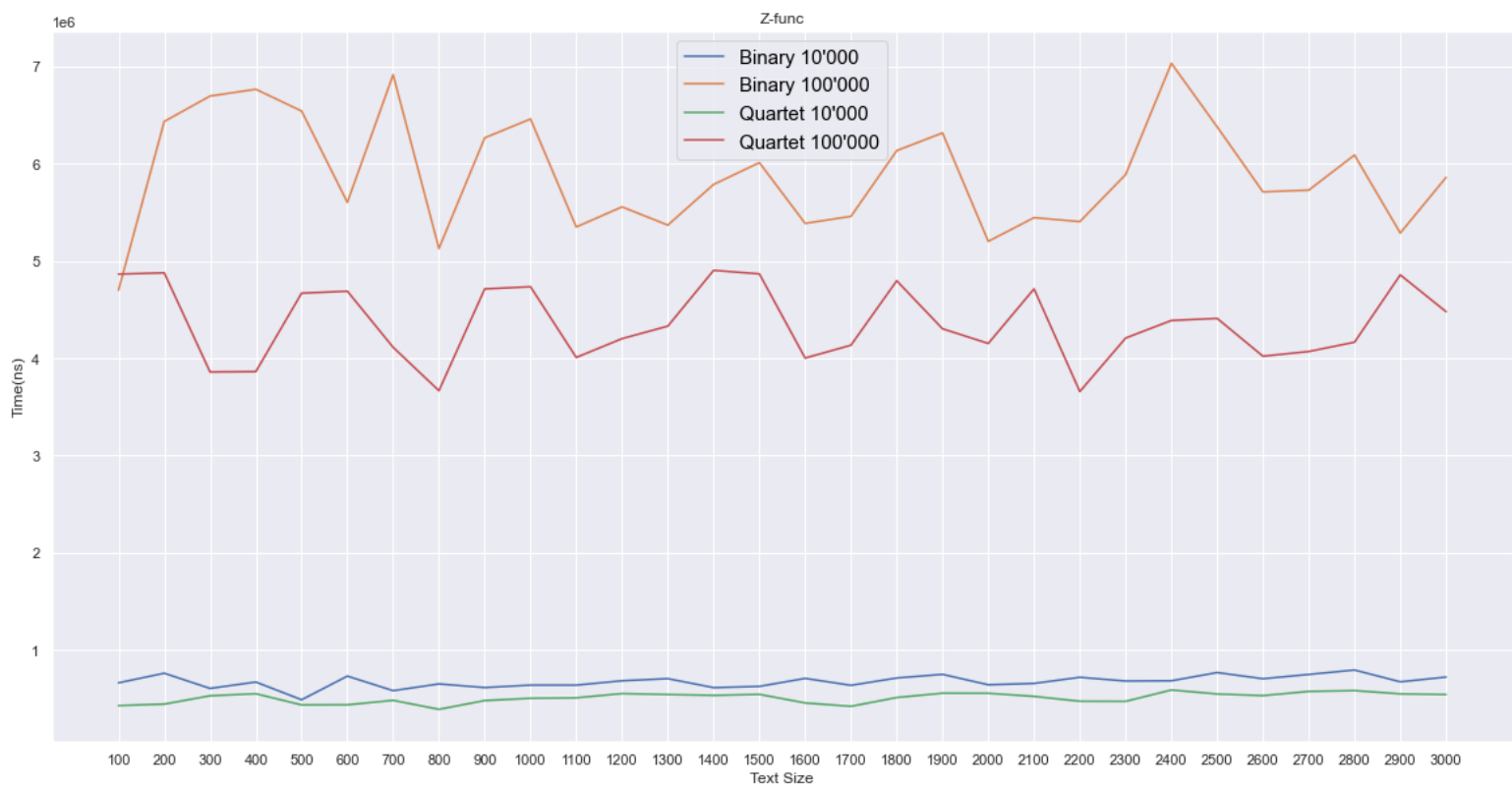




Точно также, как и в аналогичной ситуации, но без символов подстановки, однако на большем кол-ве размеров значение начало превышать за  $3+1e6$ . Разбросы стали ещё больше.



Все графики спустились ниже и начали показывать время лучше, чем с 2 знаками подстановки.

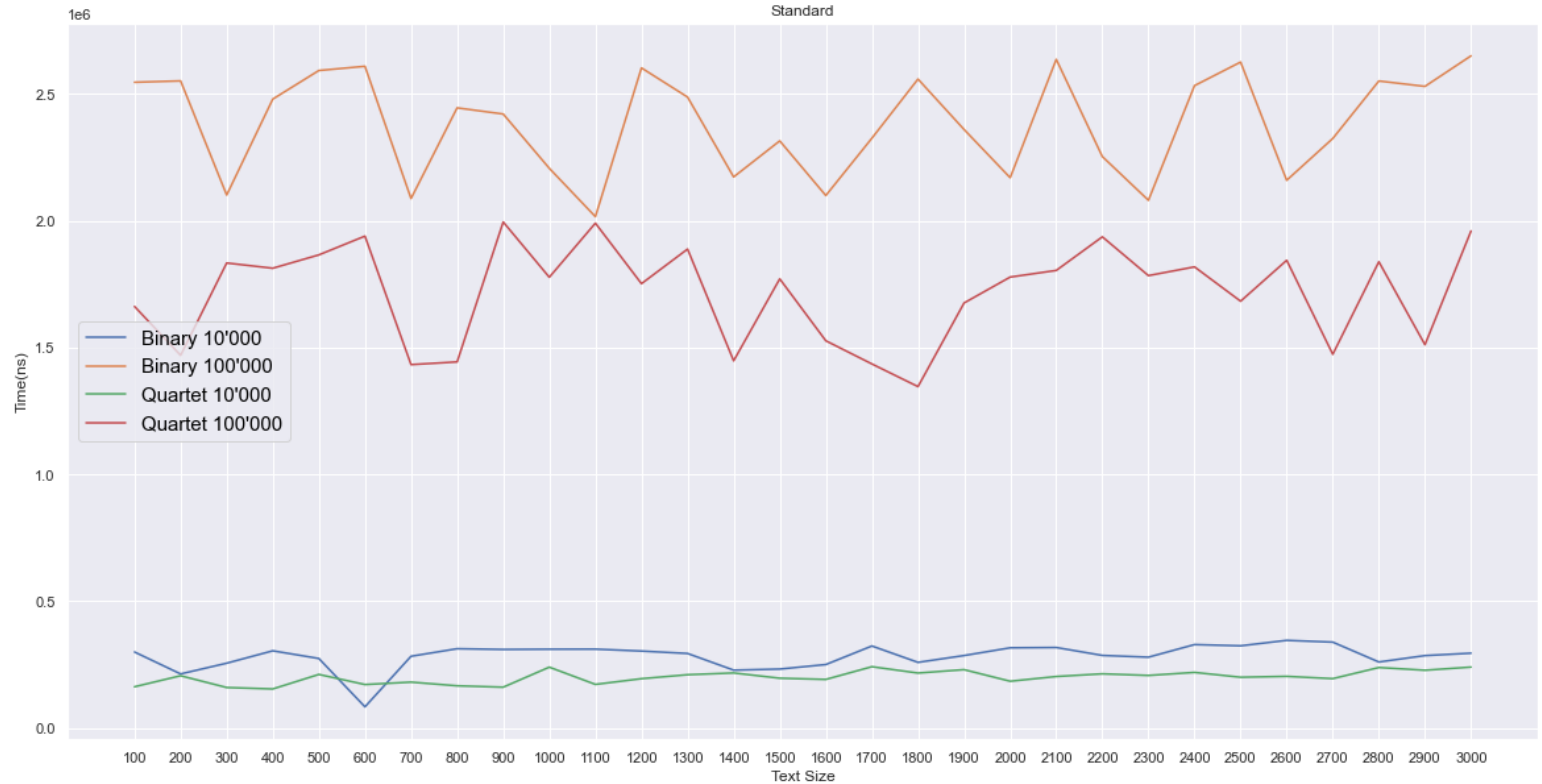


В целом, схожие результаты, как и без символа подстановки.

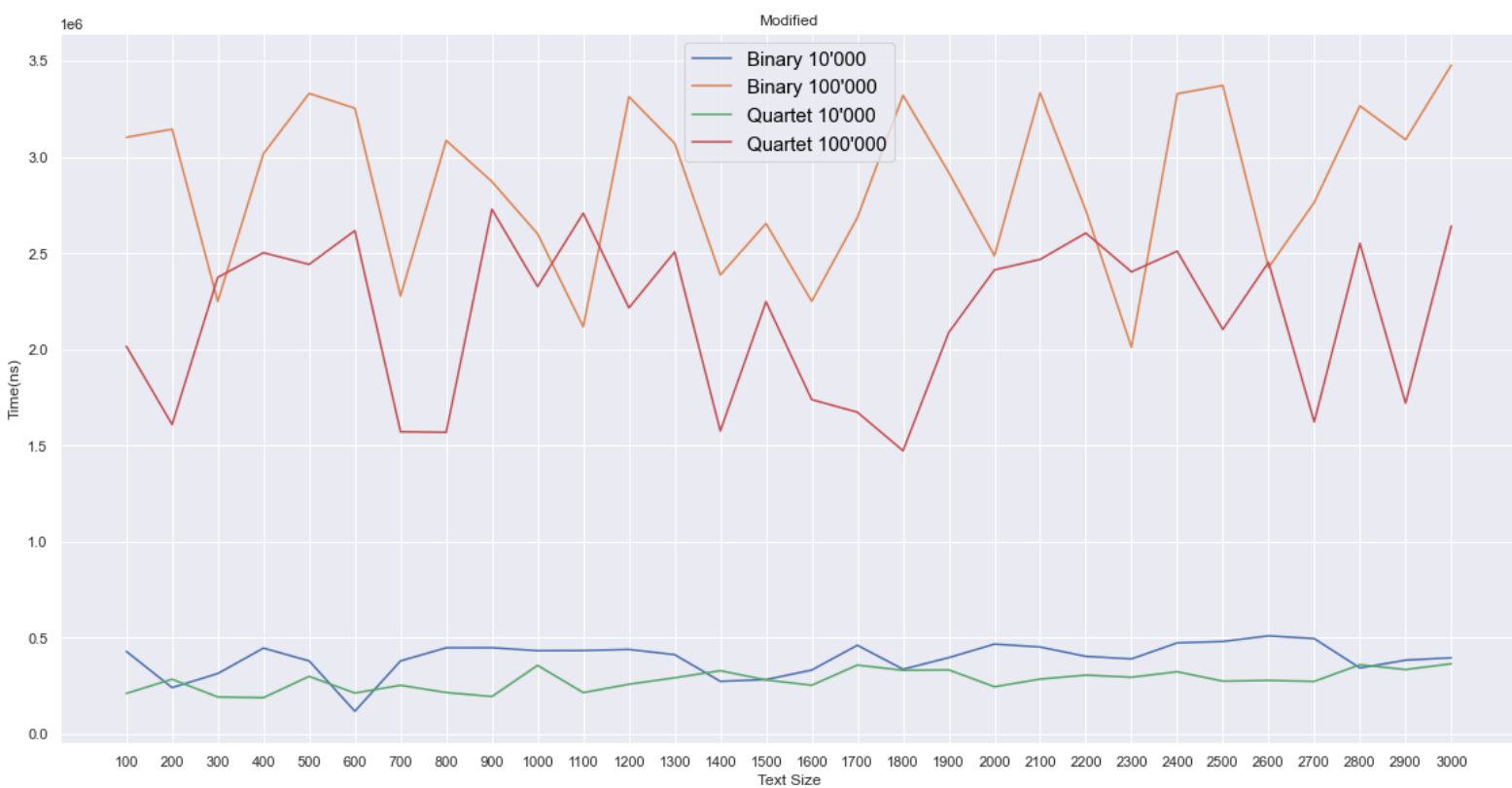
---

## 4 знака подстановки

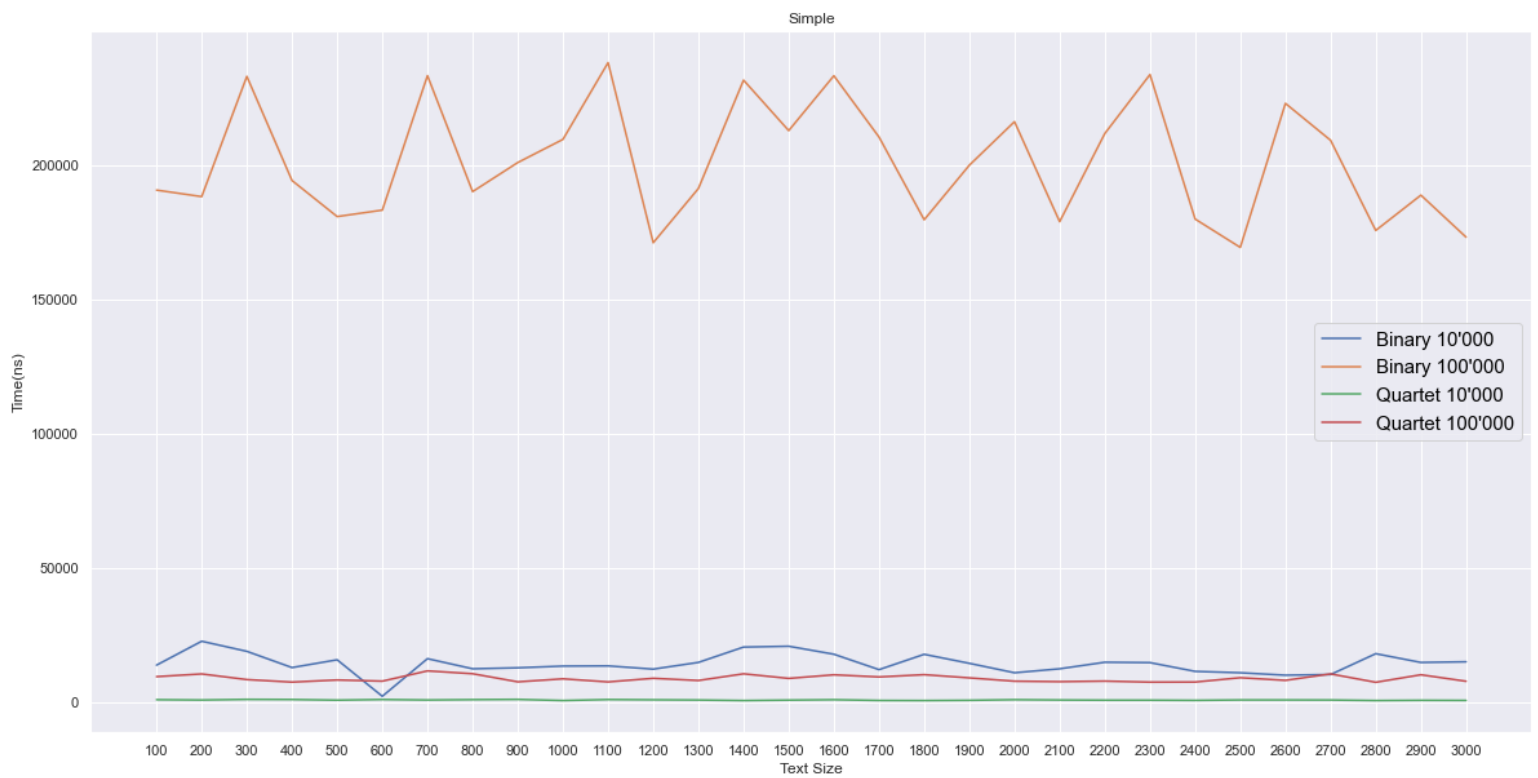
---



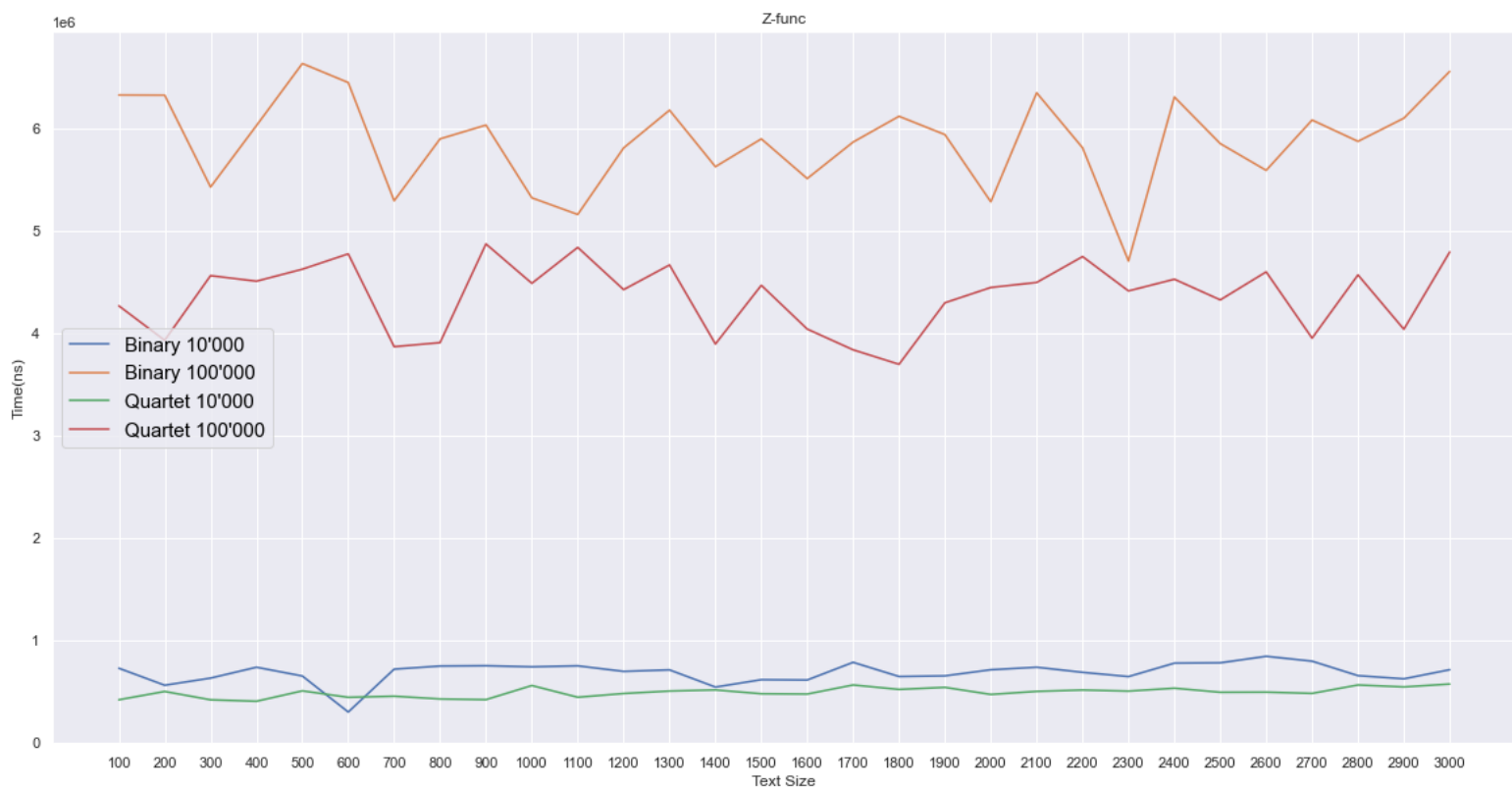
Точно также, как и в аналогичной ситуации, но без символов подстановки, однако на большем кол-ве размеров значение начало превышать за  $2.5 \cdot 10^6$ .



Точно также, как и в аналогичной ситуации, но без символов подстановки, однако на большем кол-ве размеров значение начало превышать за  $3 \cdot 10^6$ . Разбросы стали ещё больше.

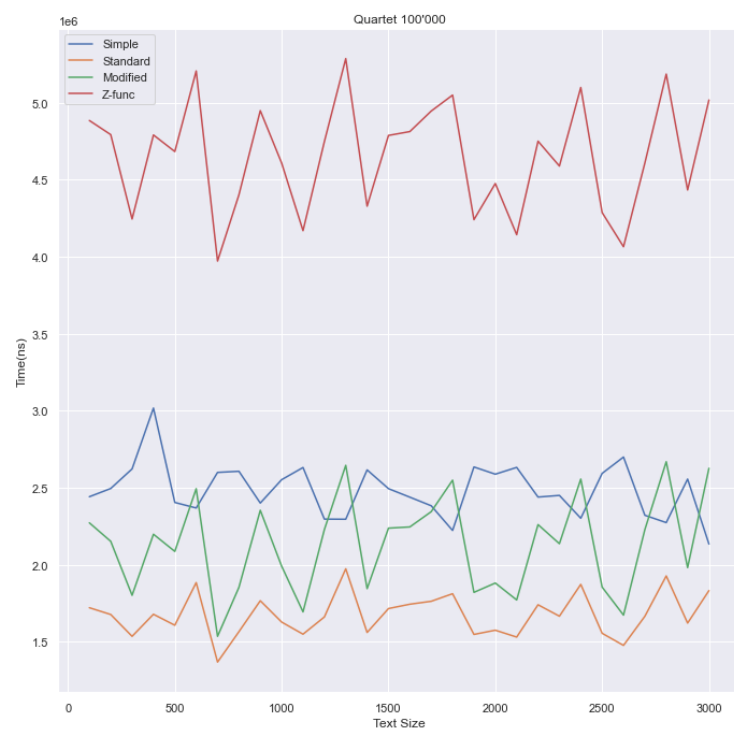
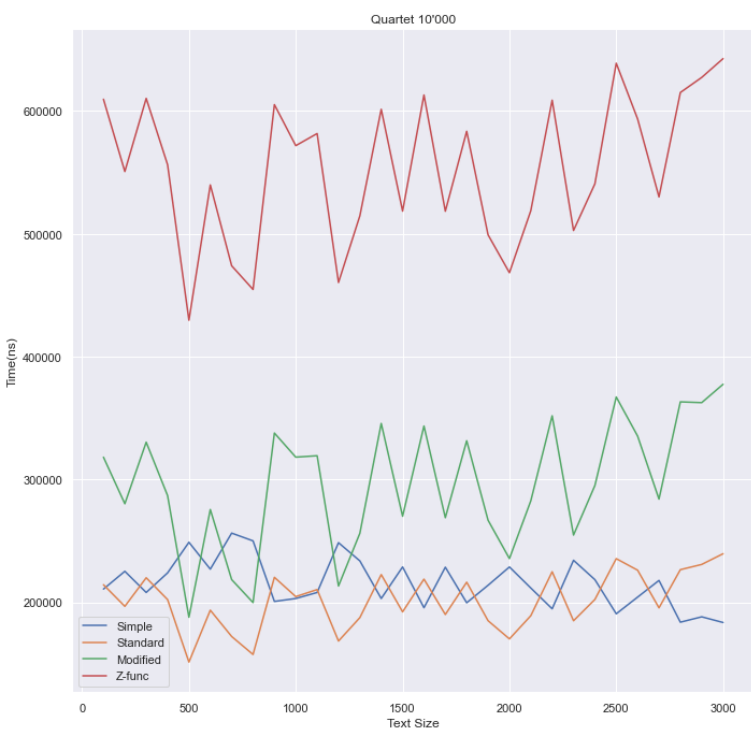
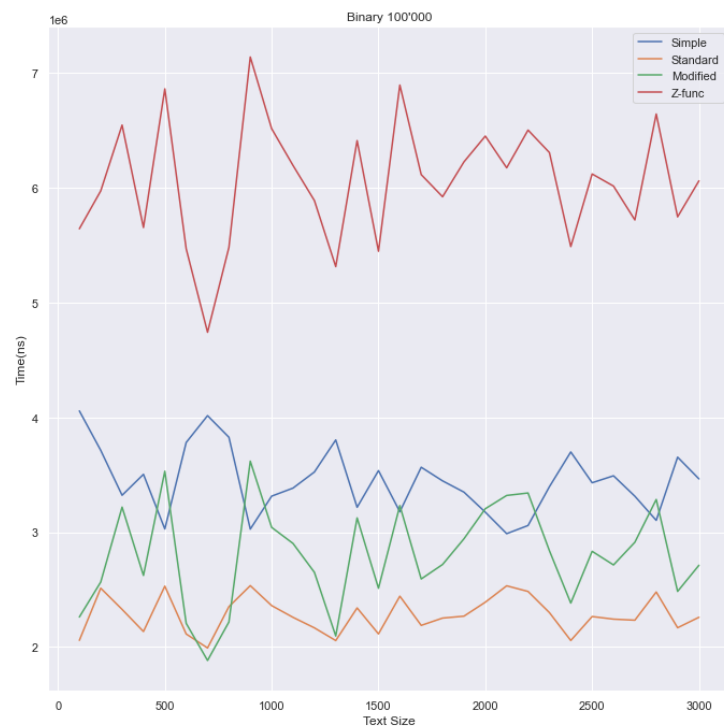
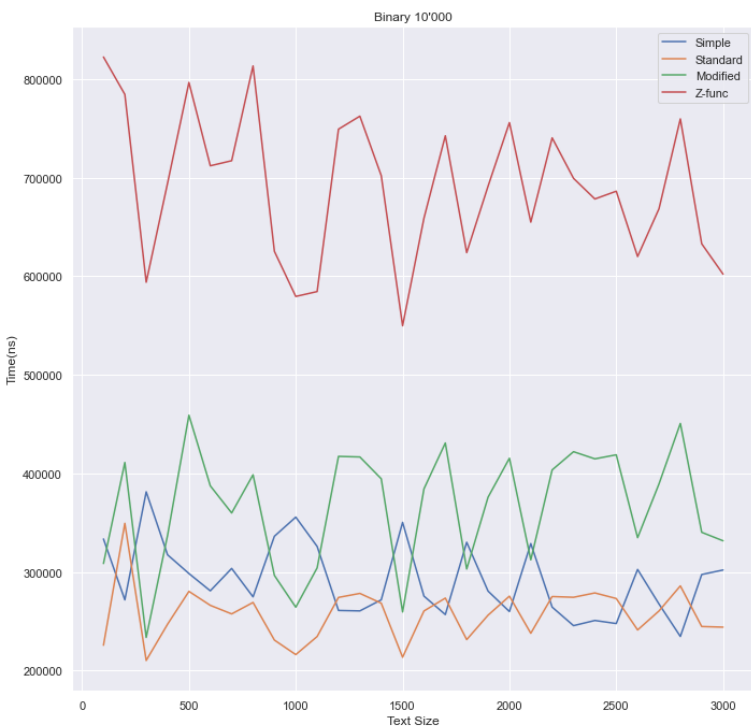


Все графики спустились ниже и начали показывать время лучше, чем с 1 знаком подстановки.



В целом, схожие результаты, как и без символа подстановки.

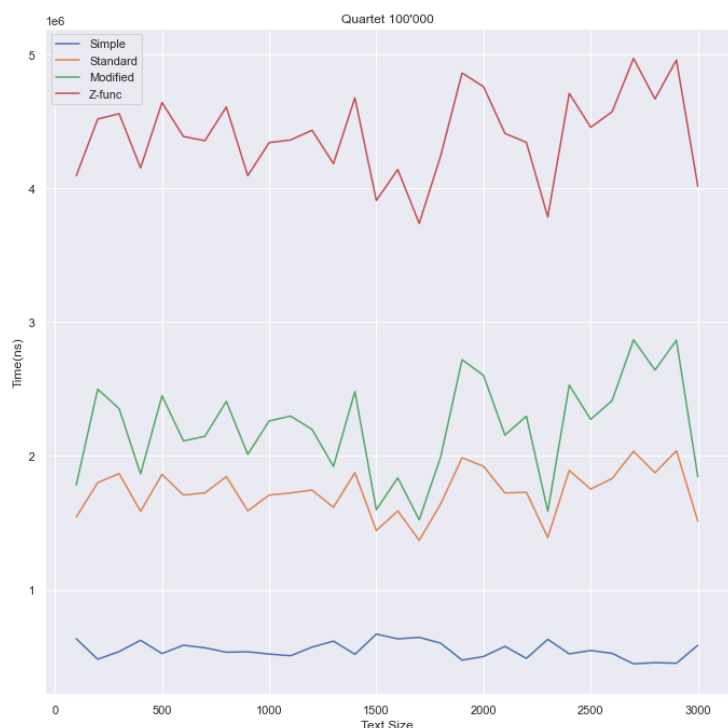
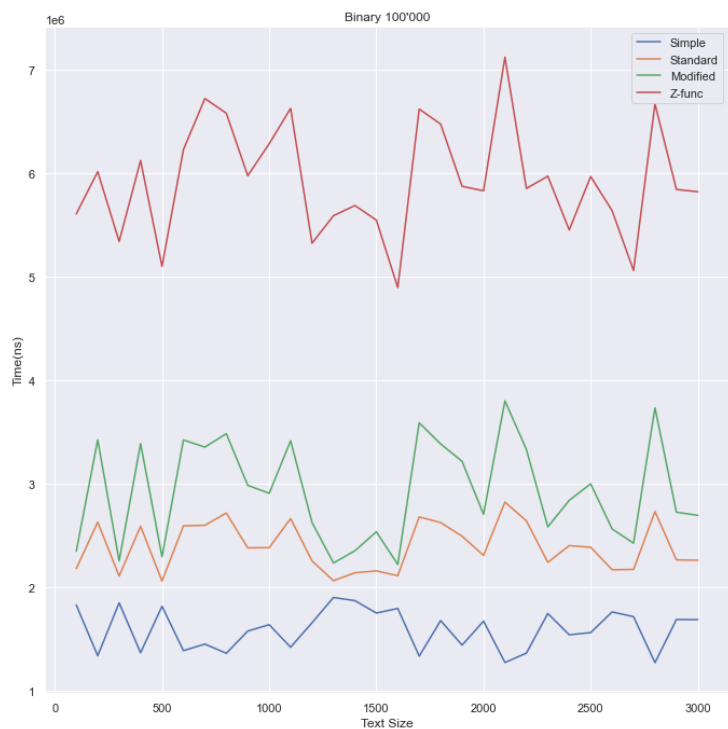
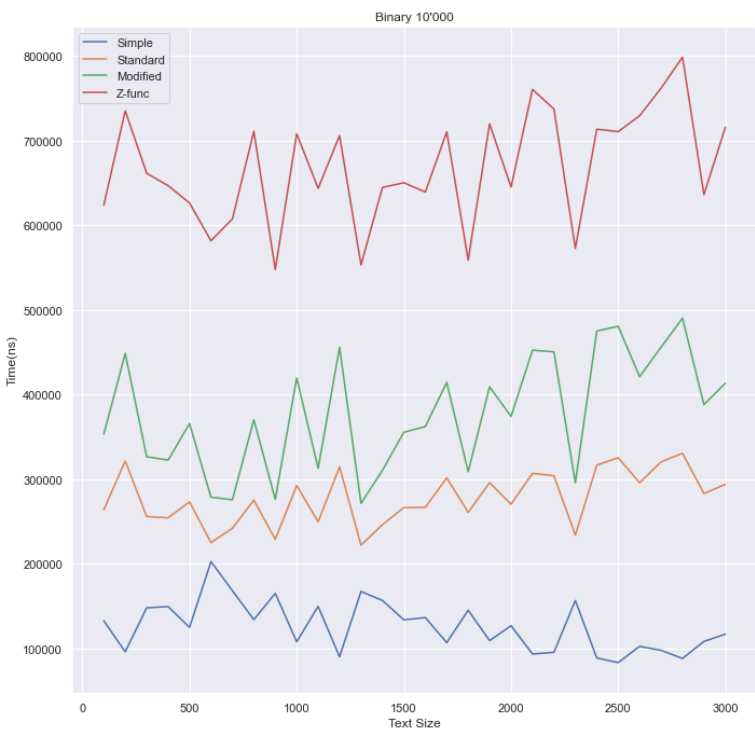
## 0 знаков подстановки



В данном случае обычный КМП работает быстрее, чем наивная реализация.



## 1 знак подстановки

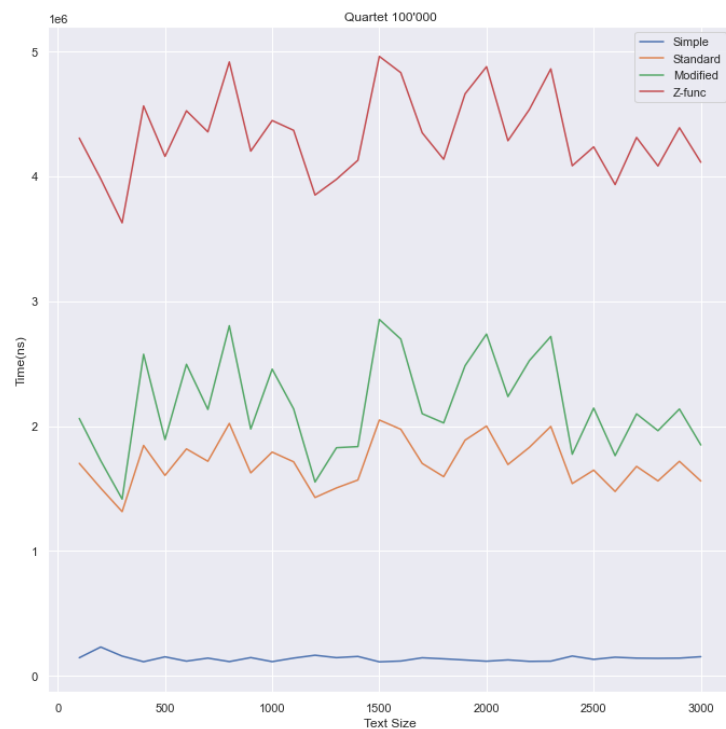
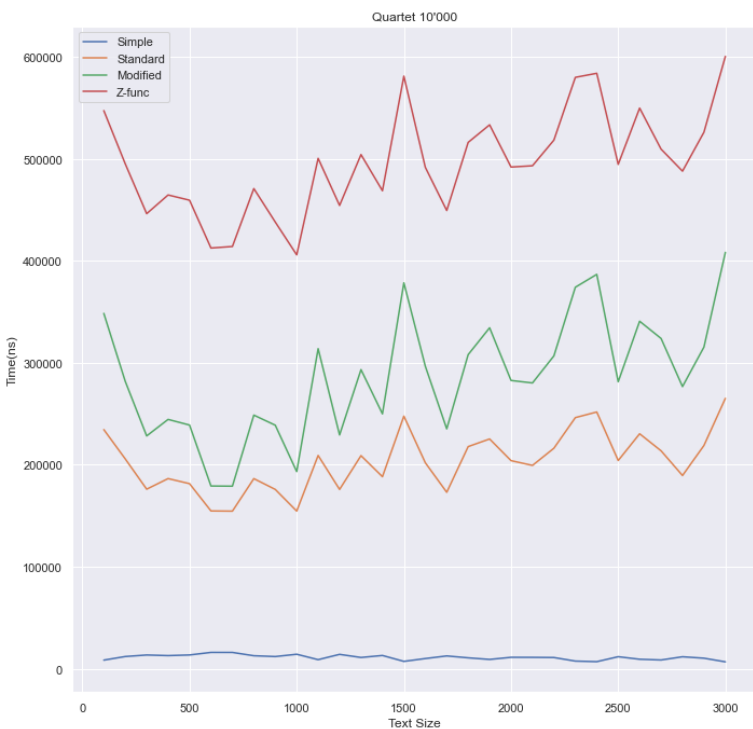
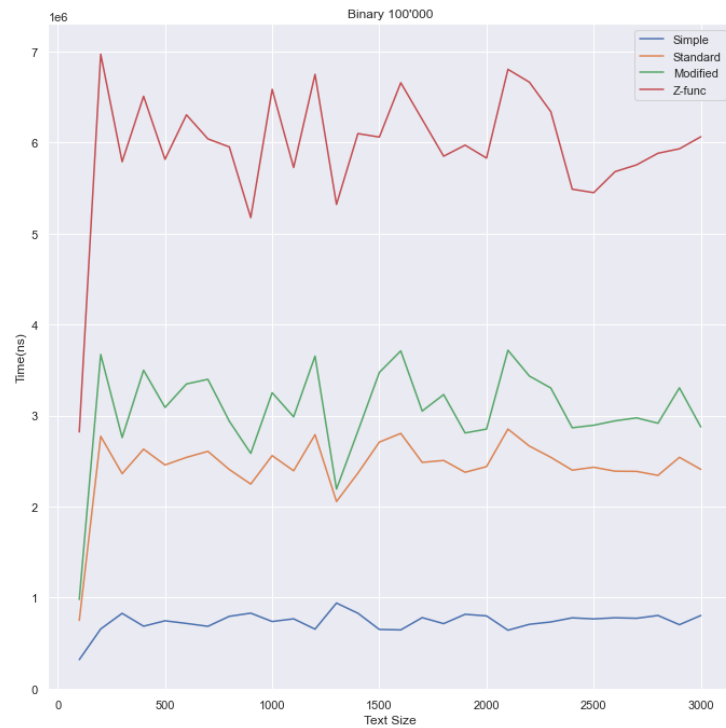
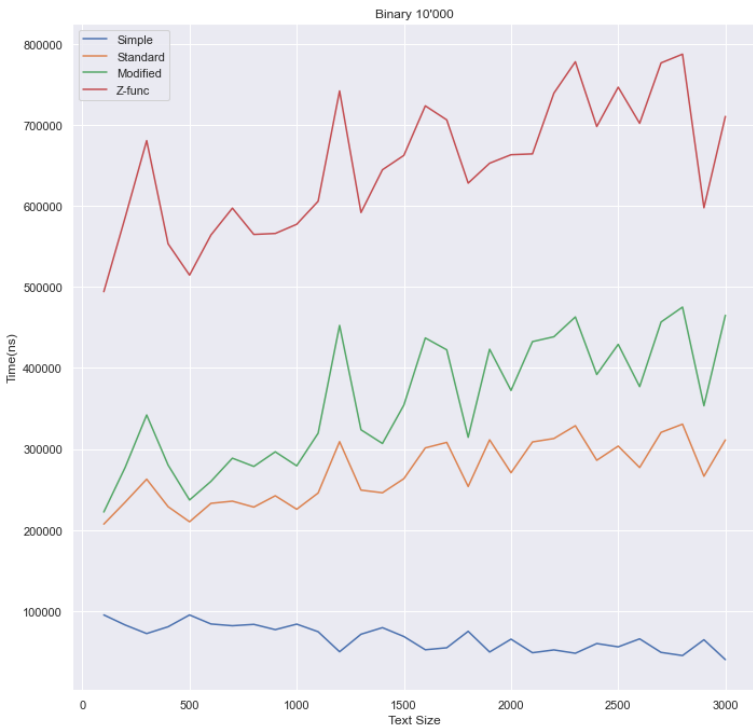


В данном случае обычный наивная реализация работает быстрее, чем все алгоритмы.

---

## 2 знака подстановки

---

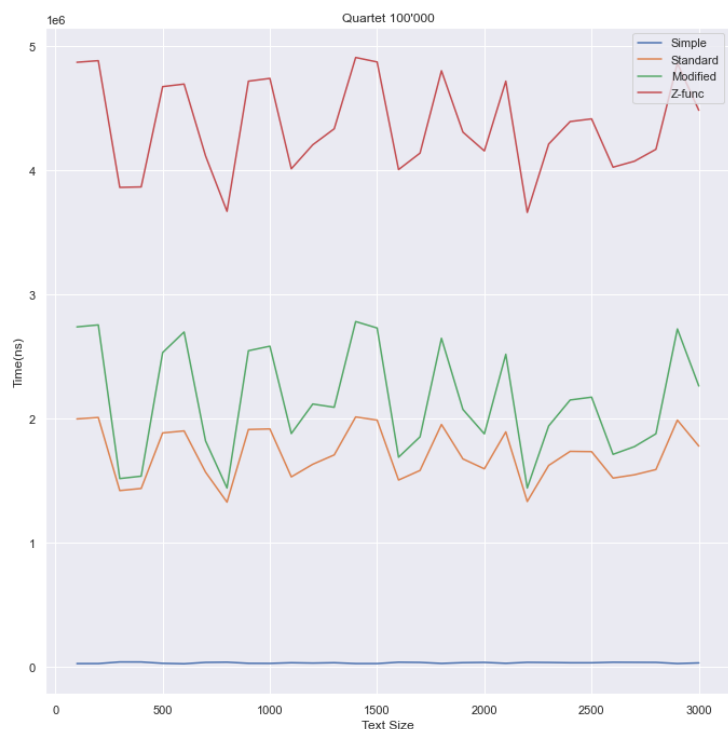
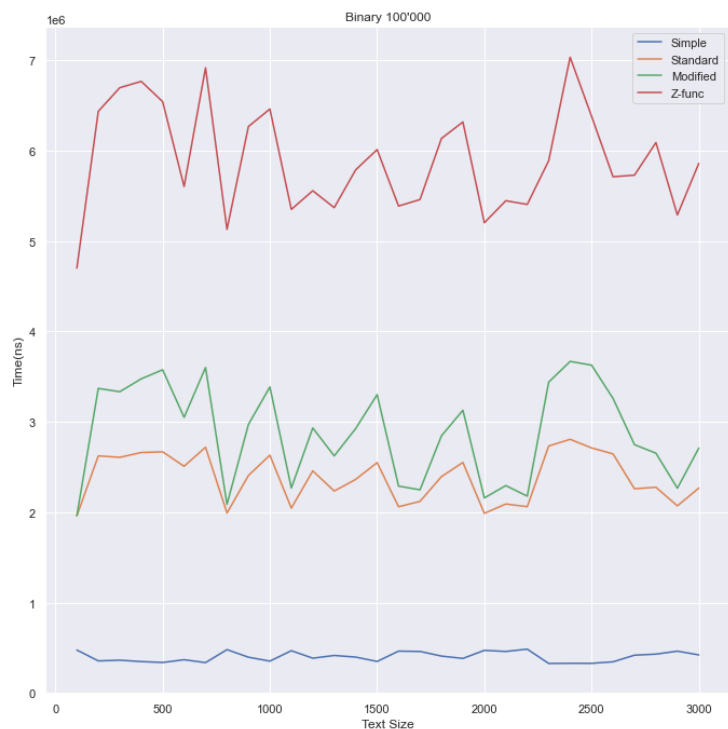


В данном случае обычный наивная реализация работает ещё быстрее, чем все алгоритмы.

---

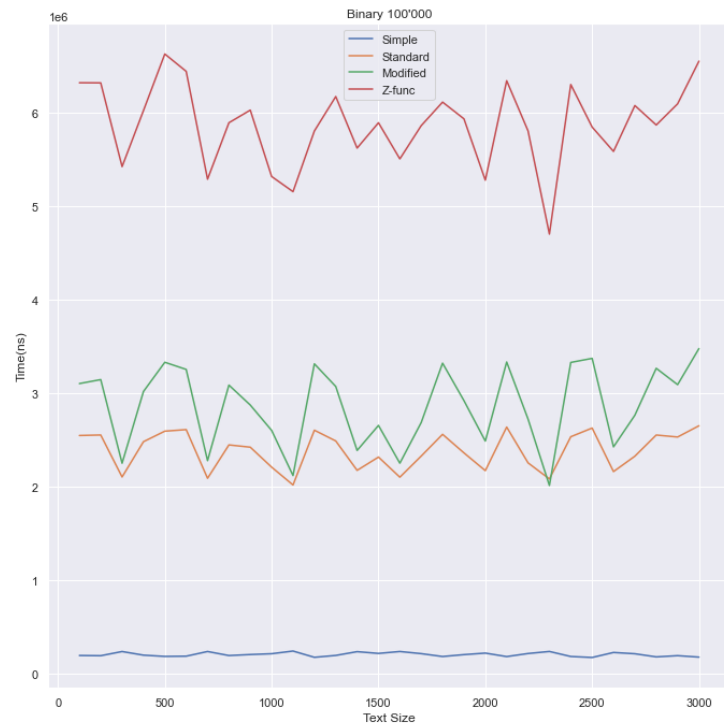
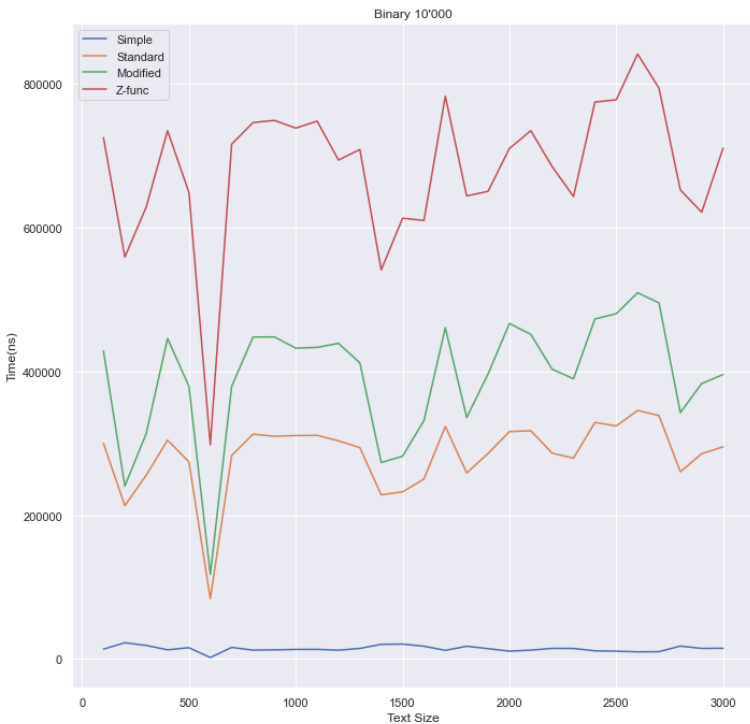
## 3 знака подстановки

---



В данном случае обычный наивная реализация работает ещё быстрее, чем все алгоритмы. Они все замедляются из-за усреднённых разнообразных комбинаций составленных шаблонов по знаку "?", в то время как наивная реализация способна быстро обрабатывать такие кейсы.

## 4 знака подстановки



В данном случае обычный наивная реализация работает ещё быстрее, чем все алгоритмы. Они все замедляются из-за усреднённых разнообразных комбинаций составленных шаблонов по знаку "?", в то время как наивная реализация способна быстро обрабатывать такие кейсы.

## **Подсчёт количества операций**

Хоть Мария Константиновна писала реализовать только что-то одно (либо графики времени, либо графики кол-ва посимвольных сравнений), тем не менее, мне было интересно дополнительно отобразить графики для кол-ва посимвольных сравнений. Они отрисованы в папке Notebooks в соответствующем файле.

## Итог

Несмотря на 10 измерений для каждого из показаний времени при отключенном интернете и снятых ненужных задач с компьютера, очень часто наблюдаются выбросы в показаниях. Они могут быть обусловлены большой нагрузкой на систему; IDE, в которой собирается и запускается программа; рандомизацией генерируемых значений текстов, позиций знаков подстановок и пр.

Из графиков можно увидеть, что время выполнения всех алгоритмов КМП линейно зависит от размера входных данных. Однако, с увеличением кол-ва символов подстановки ("?",) наивный алгоритм работает быстрее, так как он содержит оптимизированную проверку на символ "?".

Невозможно в должном образе реализовать алгоритм КМП с границами для знака "?", поэтому пришлось составлять все возможные комбинации из паттерна с этим знаком. В целом, КМП не был разработан для таких задач (работа с «подстановочными» символами), и его использование может привести к неправильной работе или нарушению асимптотической сложности.