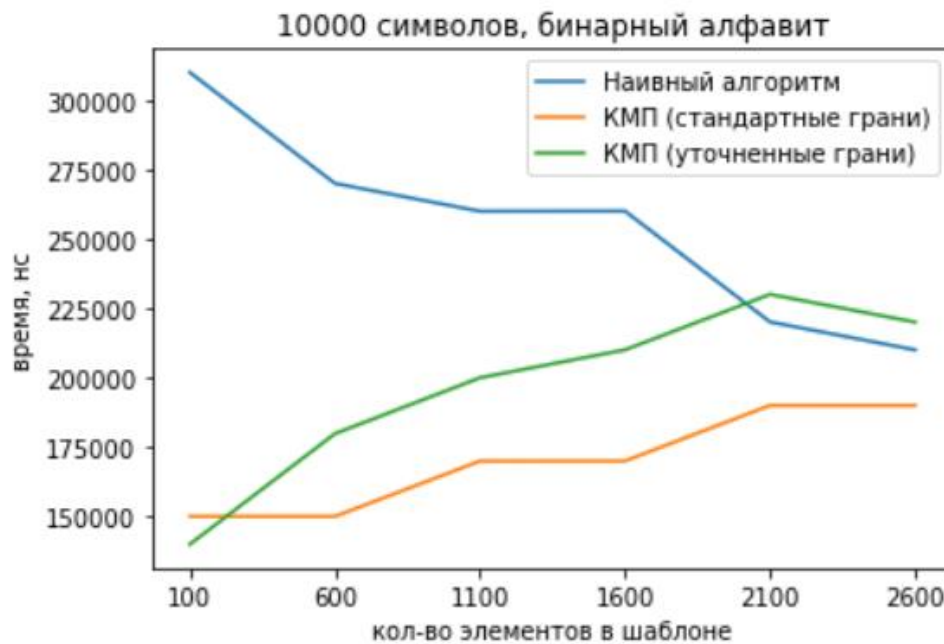
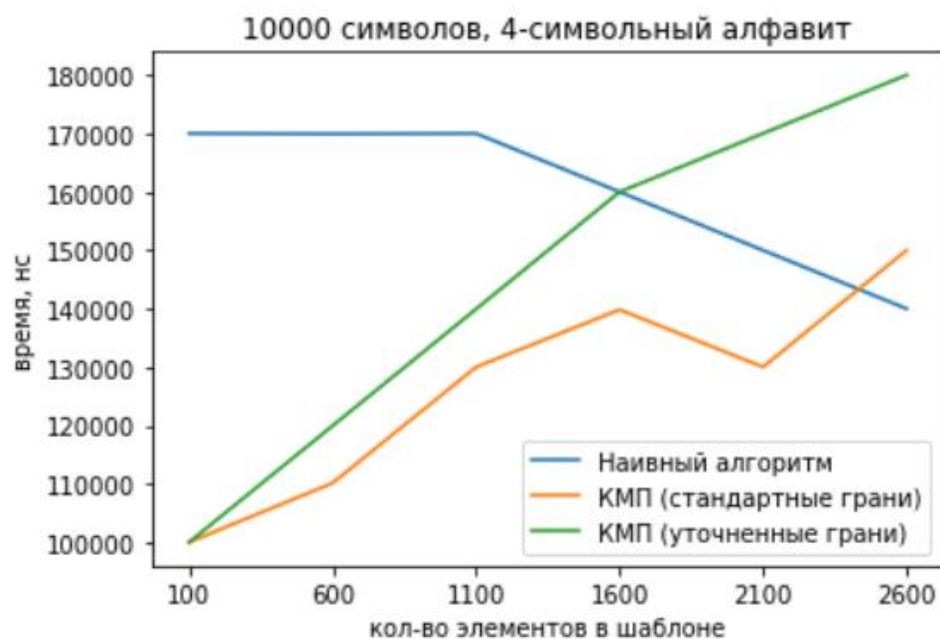


Без символов подстановки, 10000 символов, бинарный алфавит



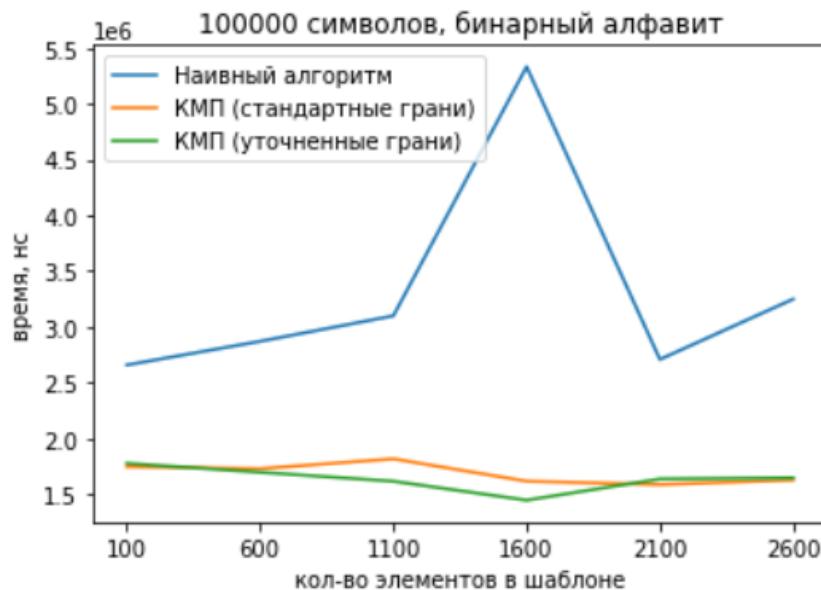
- На малых размерах подстроки наивный алгоритм работает намного дольше остальных, так как выше вероятность встретить искомую подстроку. С увеличением длины подстроки время работы наивного алгоритма также уменьшается.
- Из-за малых размеров исходной строки, КМП с уточненными границами может быть как быстрее, так и медленнее КМП со стандартными границами.

Без символов подстановки, 10000 символов, 4-символьный алфавит



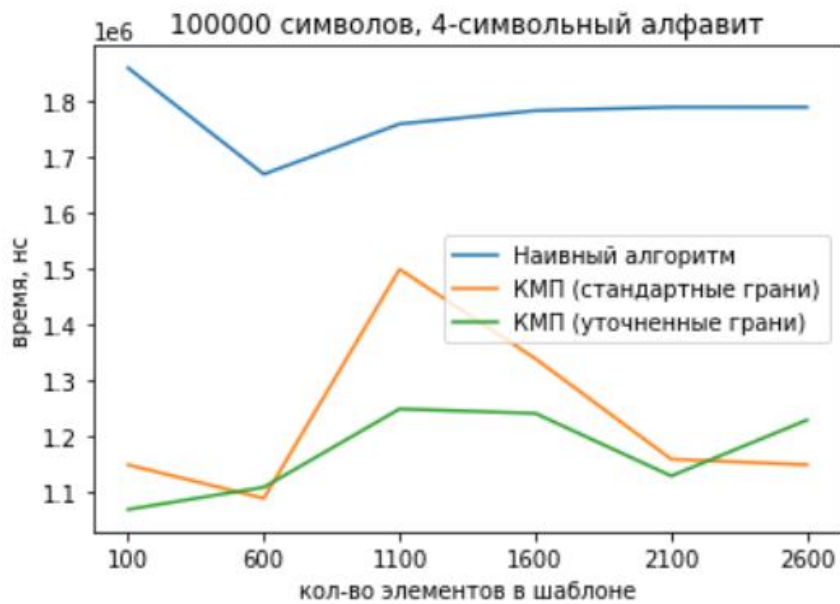
- На малых размерах подстроки наивный алгоритм работает намного дольше остальных, так как выше вероятность встретить искомую подстроку. С увеличением длины подстроки время работы наивного алгоритма также уменьшается.
- Из-за малых размеров исходной строки, КМП с уточненными границами может быть как быстрее, так и медленнее КМП со стандартными границами.

Без символов подстановки, 100000 символов, бинарный алфавит



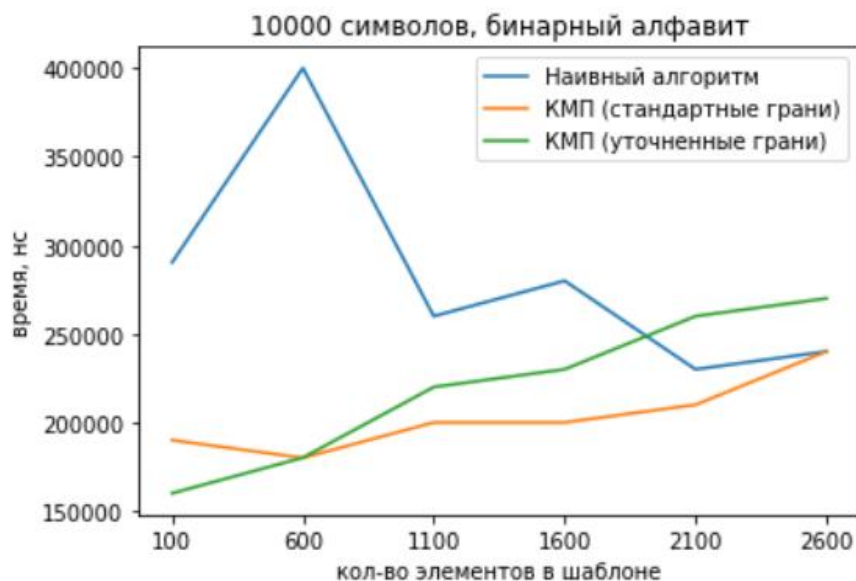
- На малых размерах подстроки наивный алгоритм работает намного дольше остальных, так как выше вероятность встретить искомую подстроку. С увеличением длины подстроки время работы наивного алгоритма не сильно уменьшается, так как искомая строка в разы больше, поэтому вероятность встретить подстроку все равно остается немалой.
- На исходном размере строки в 100к символов уже заметно, как в среднем КМП с уточненными границами работают быстрее стандартного КМП.

Без символов подстановки, 100000 символов, 4-символьный алфавит



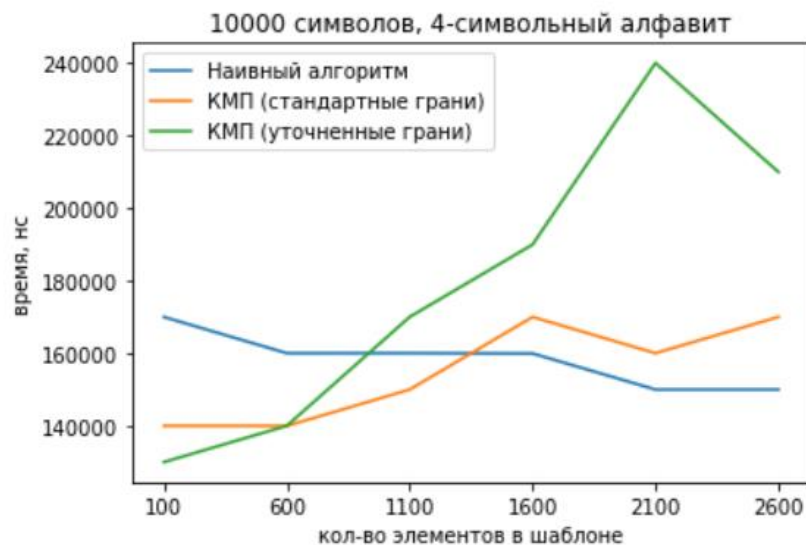
- На малых подстроки размерах наивный алгоритм работает намного дольше остальных, так как выше вероятность встретить искомую подстроку. С увеличением длины подстроки время работы наивного алгоритма не сильно уменьшается, так как искомая строка в разы больше, поэтому вероятность встретить подстроку все равно остается немалой.
- На исходном размере строки в 100к символов уже заметно, как в среднем КМП с уточненными гранями работают быстрее стандартного КМП.

С 1 символом подстановки, 10000 символов, бинарный алфавит



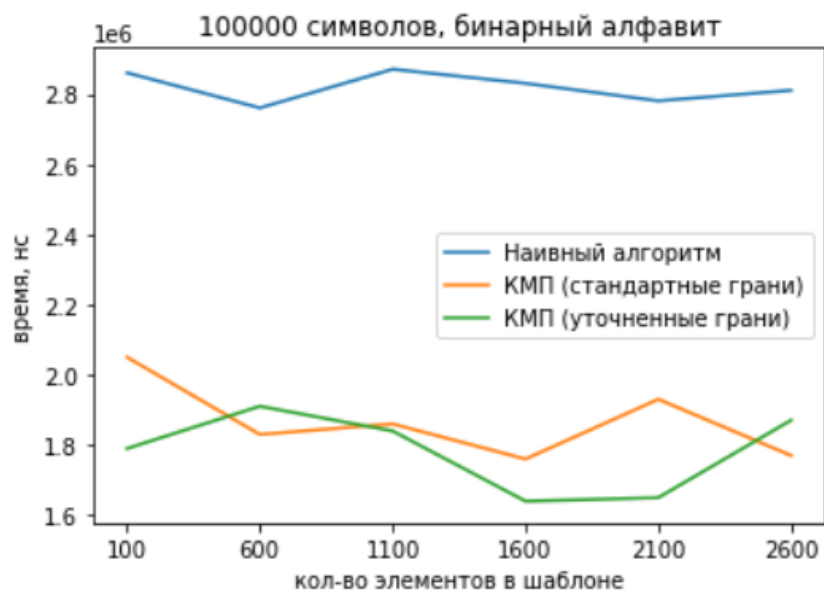
- На малых размерах подстроки наивный алгоритм работает намного дольше остальных, так как выше вероятность встретить искомую подстроку. С увеличением длины подстроки время работы наивного алгоритма также уменьшается.
- Из-за малых размеров исходной строки, КМП с уточненными границами может быть как быстрее, так и медленнее КМП со стандартными границами.

С 1 символом подстановки, 10000 символов, 4-символьный алфавит



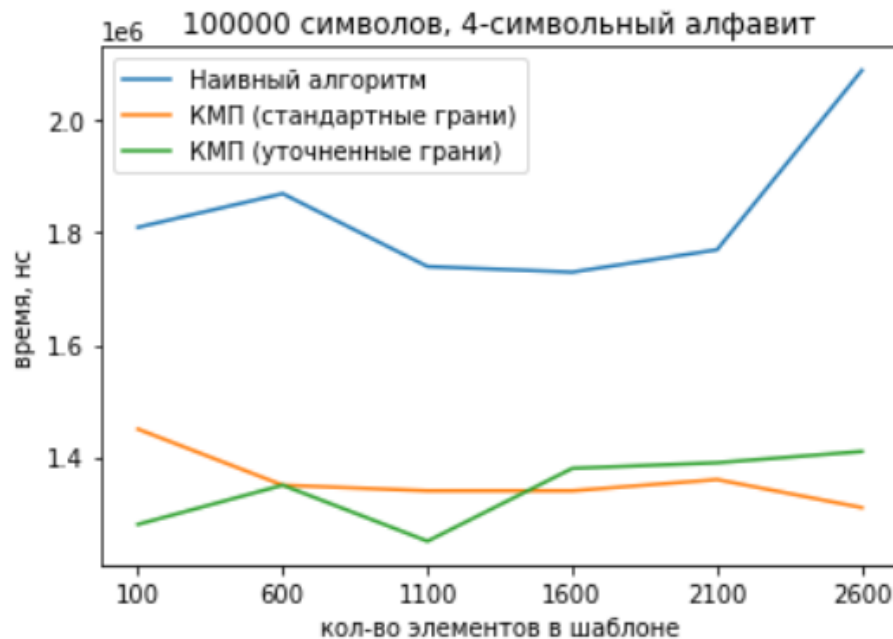
- Во время проведения испытаний на малых размерах исходной строки в некоторых случаях наивный алгоритм может работать даже быстрее остальных. Но, как мы увидим на более длинных строках, это просто погрешности измерений.

С 1 символом подстановки, 100000 символов, бинарный алфавит



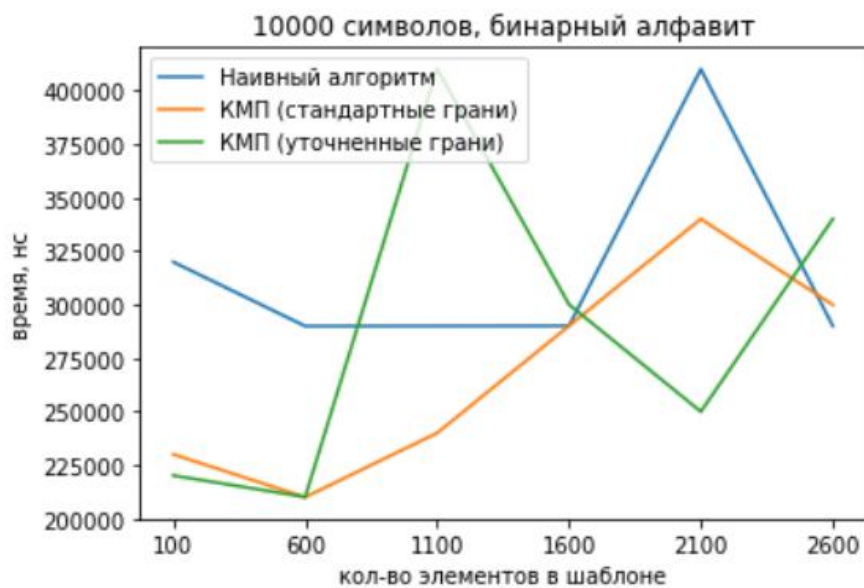
- Наивный алгоритм работает в разы дольше двух других. КМП с уточненными границами, в свою очередь, работает быстрее всех.
- Также можно заметить, что время работы алгоритмов увеличилось из-за появления символа подстановки, которое повлекло дополнительные сравнения в алгоритмах.

С 1 символом подстановки, 100000 символов, 4-символьный алфавит



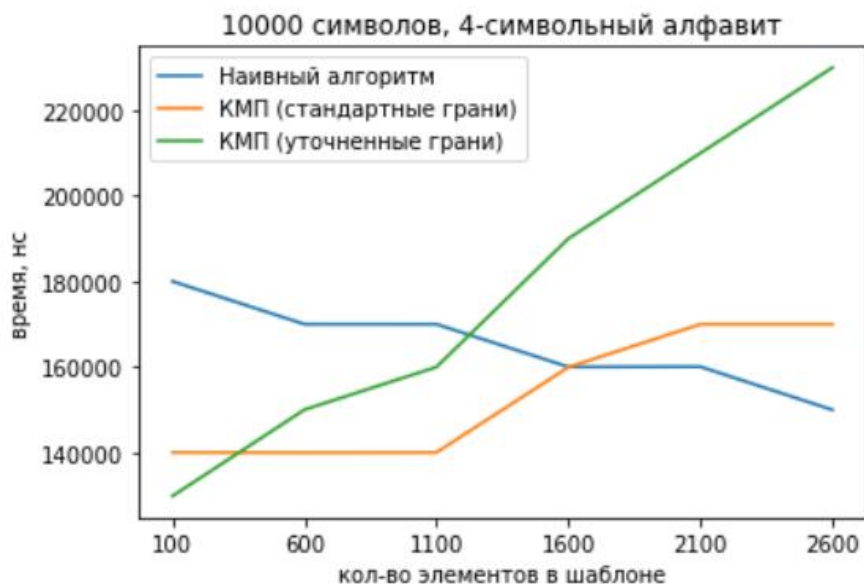
- Наивный алгоритм работает в разы дольше двух других.
- Также можно заметить, что время работы алгоритмов увеличилось из-за появления символа подстановки, которое повлекло дополнительные сравнения в алгоритмах.

С 2 символами подстановки, 10000 символов, бинарный алфавит



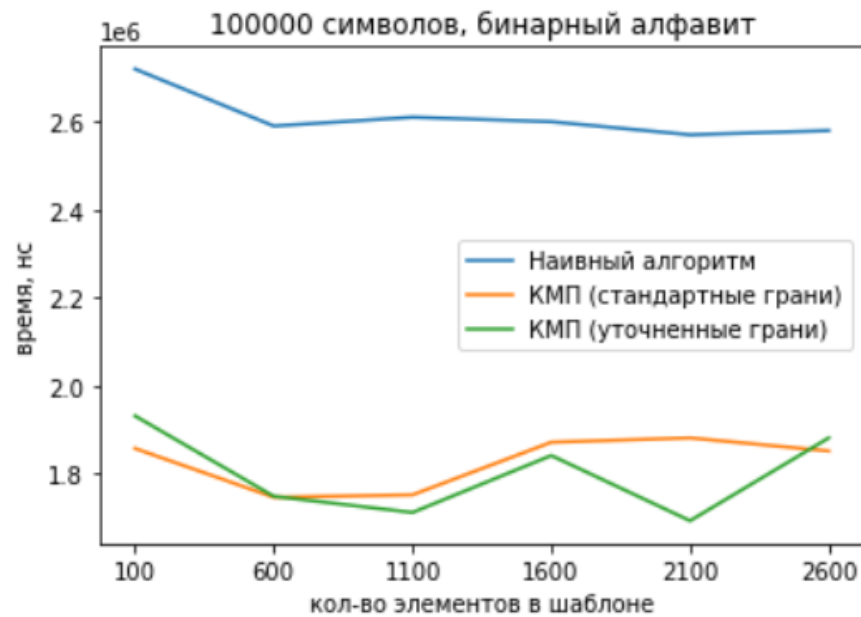
- Во время проведения испытаний на малых размерах исходной строки в некоторых случаях наивный алгоритм может работать даже быстрее остальных. Но, как мы увидим на более длинных строках, это просто погрешности измерений.

С 2 символами подстановки, 10000 символов, 4-символьный алфавит



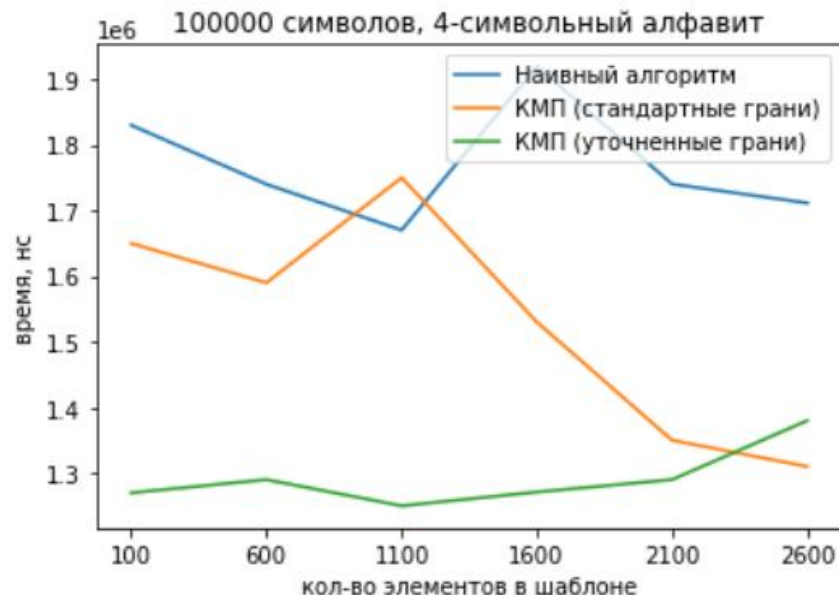
- Во время проведения испытаний на малых размерах исходной строки в некоторых случаях наивный алгоритм может работать даже быстрее остальных. Но, как мы увидим на более длинных строках, это просто погрешности измерений.

С 2 символами подстановки, 100000 символов, бинарный алфавит



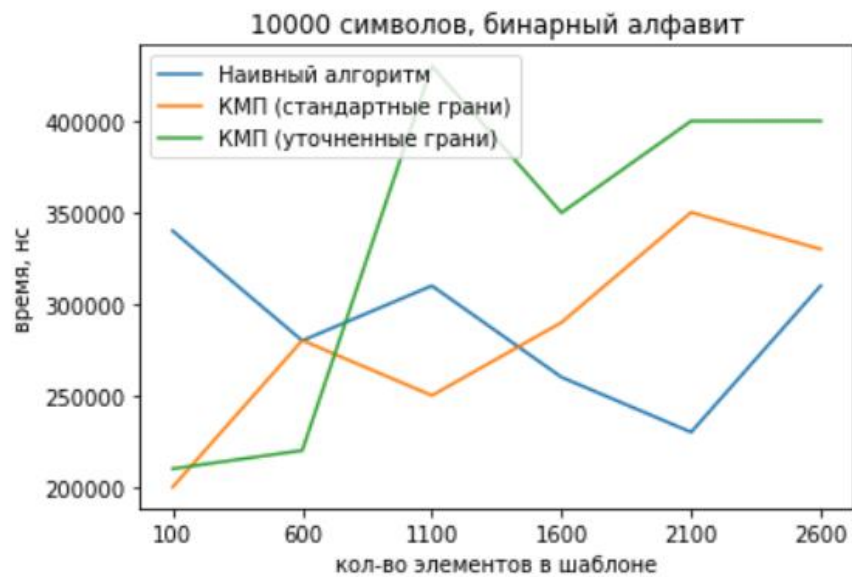
- Наивный алгоритм работает в разы дольше двух других. КМП с уточненными границами, в свою очередь, работает быстрее всех.
- Также можно заметить, что время работы алгоритмов увеличилось из-за появления символов подстановки, которые повлекли дополнительные сравнения в алгоритмах.

С 2 символами подстановки, 100000 символов, 4-символьный алфавит



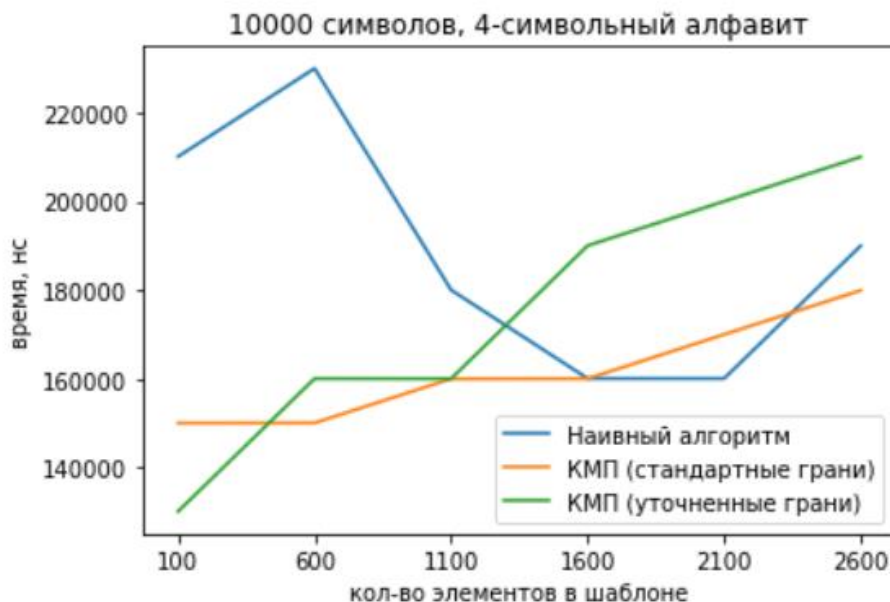
- Из графика можно увидеть, как в некоторых случаях стандартный КМП может быть не особо эффективным, однако КМП с уточненными границами, в свою очередь, все также работает очень быстро.

С 3 символами подстановки, 10000 символов, бинарный алфавит



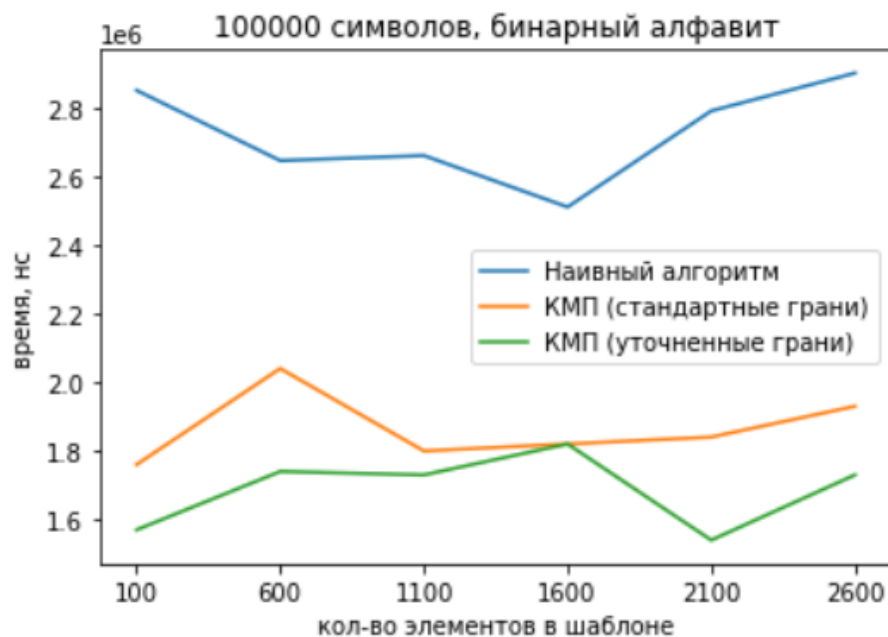
- Во время проведения испытаний на малых размерах исходной строки в некоторых случаях наивный алгоритм может работать даже быстрее остальных. Но, как мы увидим на более длинных строках, это просто погрешности измерений.

С 3 символами подстановки, 10000 символов, 4-символьный алфавит



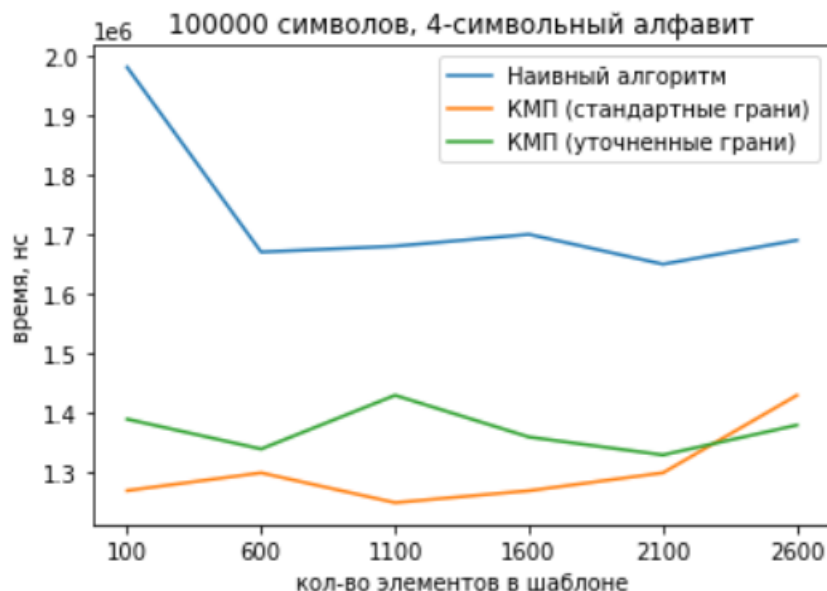
- Во время проведения испытаний на малых размерах исходной строки в некоторых случаях наивный алгоритм может работать даже быстрее остальных. Но, как мы увидим на более длинных строках, это просто погрешности измерений.

С 3 символами подстановки, 100000 символов, бинарный алфавит



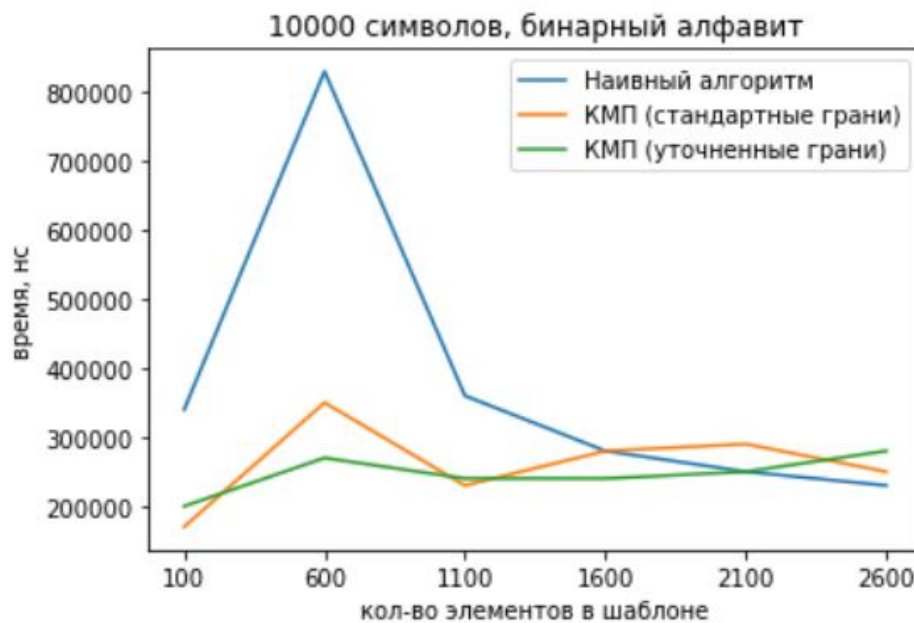
- Наивный алгоритм работает в разы дольше двух других. КМП с уточненными границами, в свою очередь, работает быстрее всех.
- Также можно заметить, что время работы алгоритмов увеличилось из-за появления символов подстановки, которые повлекли дополнительные сравнения в алгоритмах.

С 3 символами подстановки, 100000 символов, 4-символьный алфавит



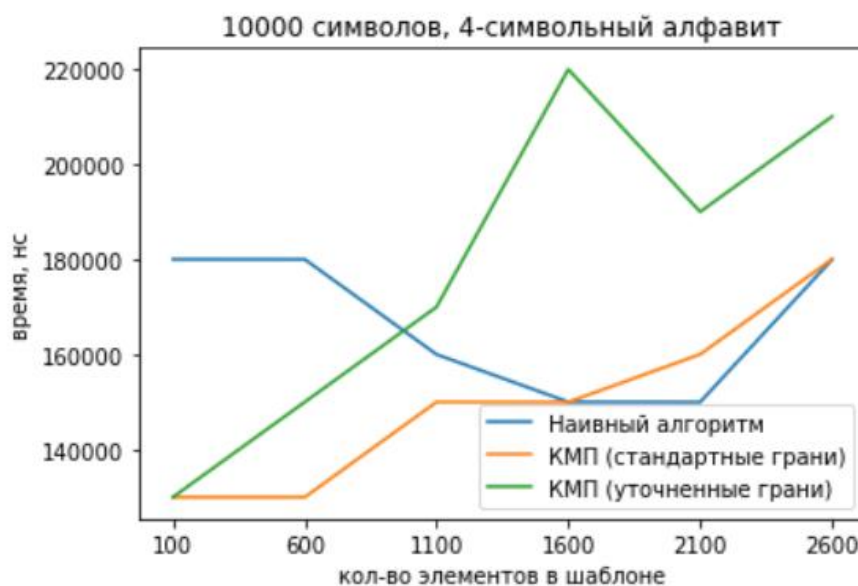
- Наивный алгоритм работает в разы дольше двух других.
- Также можно заметить, что время работы алгоритмов увеличилось из-за появления символов подстановки, которые повлекли дополнительные сравнения в алгоритмах.

С 4 символами подстановки, 10000 символов, бинарный алфавит



- На малых размерах подстроки наивный алгоритм работает намного дольше остальных, так как выше вероятность встретить искомую подстроку. С увеличением длины подстроки время работы наивного алгоритма также уменьшается.
- Из-за малых размеров исходной строки, КМП с уточненными границами может быть как быстрее, так и медленнее КМП со стандартными границами.

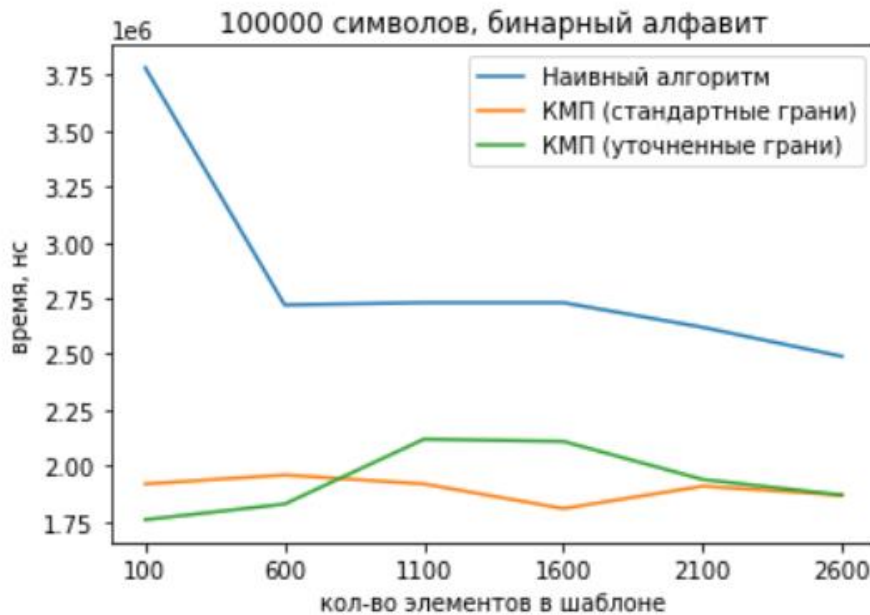
С 4 символами подстановки, 10000 символов, 4-символьный алфавит



- Во время проведения испытаний на малых размерах исходной строки в некоторых случаях наивный алгоритм может работать даже быстрее

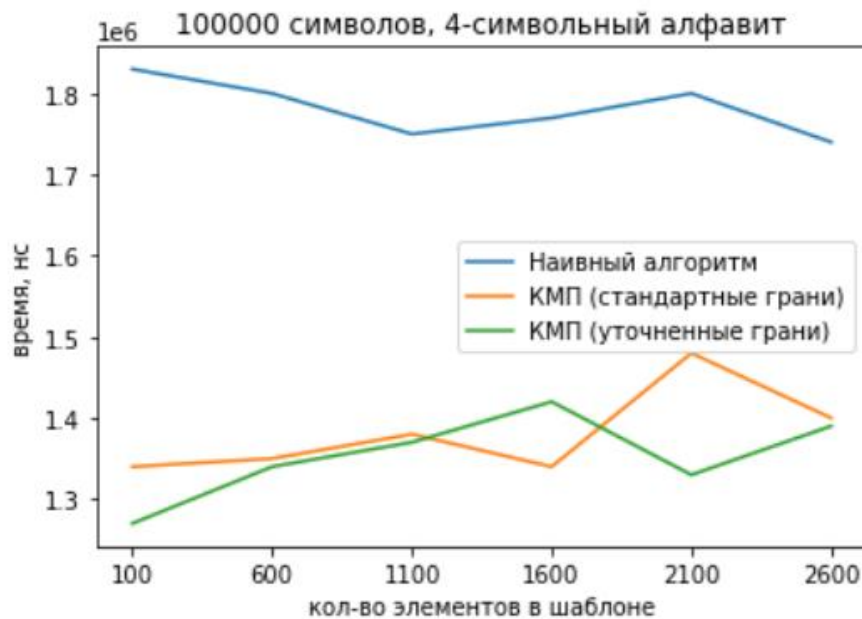
остальных. Но, как мы увидим на более длинных строках, это просто погрешности измерений.

С 4 символами подстановки, 100000 символов, бинарный алфавит



- Наивный алгоритм работает в разы дольше двух других.
- Также можно заметить, что время работы алгоритмов увеличилось из-за появления символов подстановки, которые повлекли за собой дополнительные сравнения в алгоритмах.

С 4 символами подстановки, 100000 символов, 4-символьный алфавит



- Наивный алгоритм работает в разы дольше двух других. КМП с уточненными границами, в свою очередь, работает быстрее всех.

- Также можно заметить, что время работы алгоритмов увеличилось из-за появления символов подстановки, которые повлекли дополнительные сравнения в алгоритмах.

Выводы

В ходе работы экспериментальным путем была доказана эффективность работы 3 алгоритмов поиска вхождений строки-шаблона в тексте. Наивный алгоритм оказался наиболее медленным из всех. Алгоритм КМП с уточненными границами в среднем показал наибольшую эффективность. Также стоит отметить, что с увеличением числа символов подстановок увеличивается общее время работы всех алгоритмов.