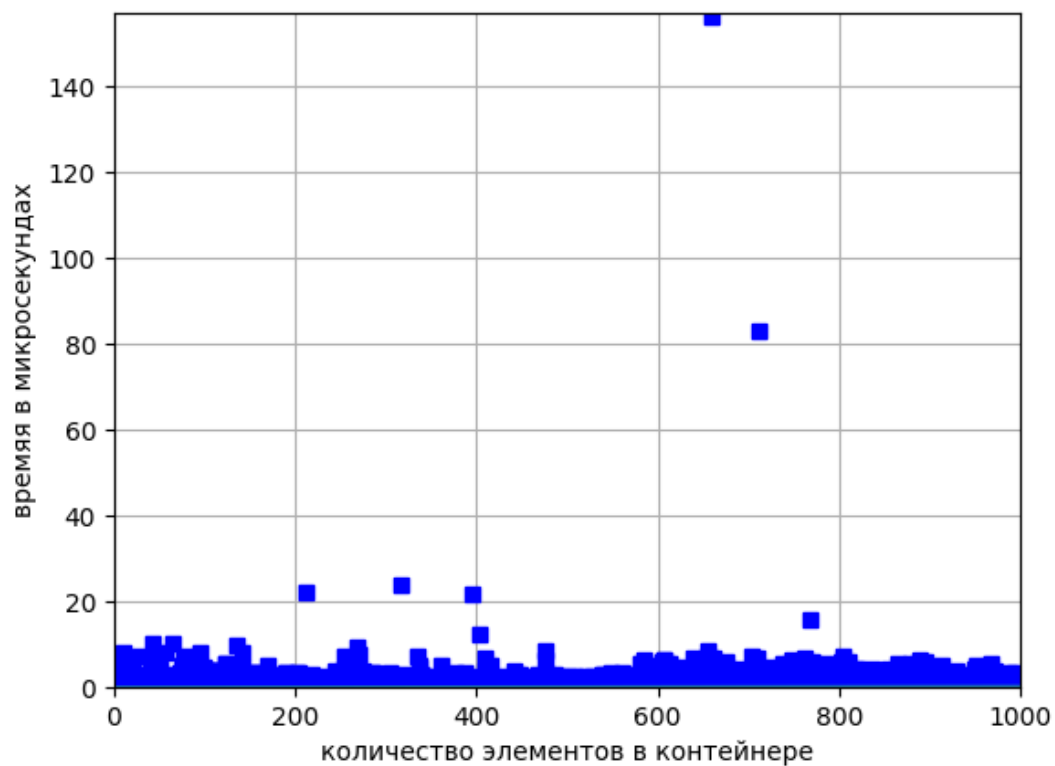
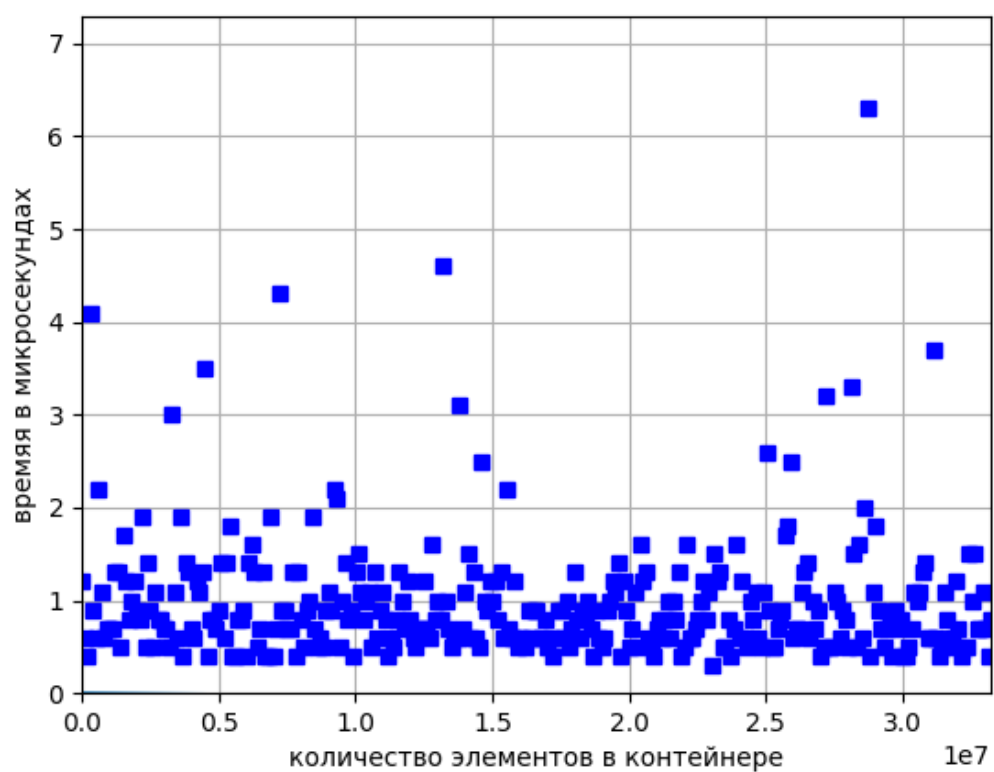


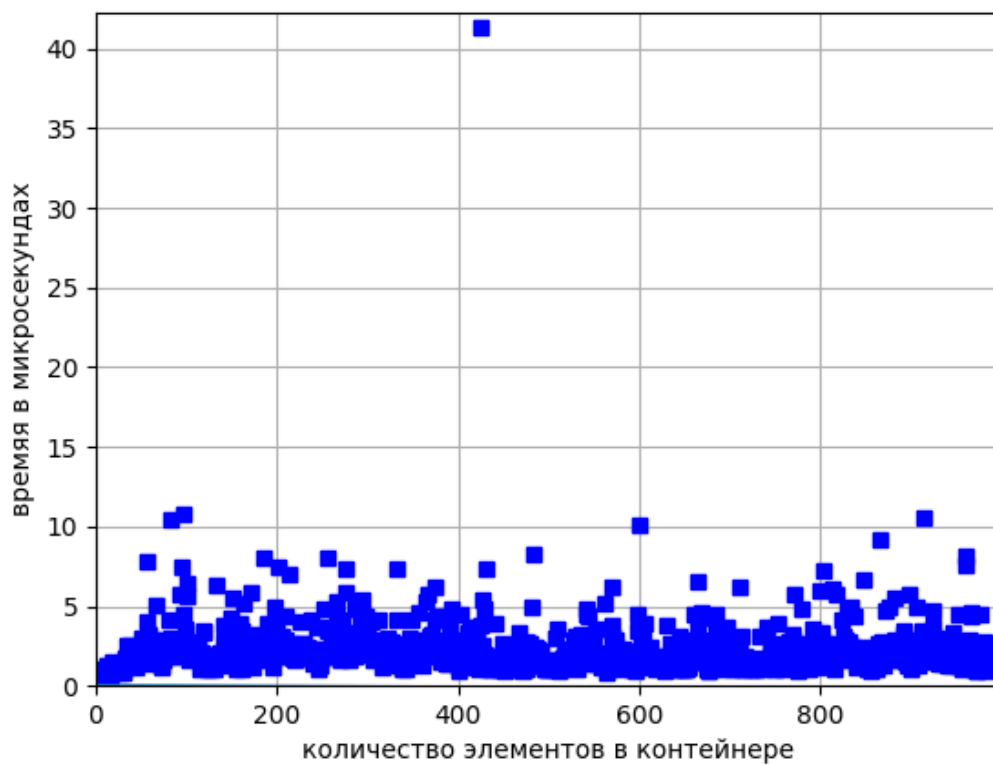
Вектор с максимальным количеством элементов = 3.3×10^7 (push_back())



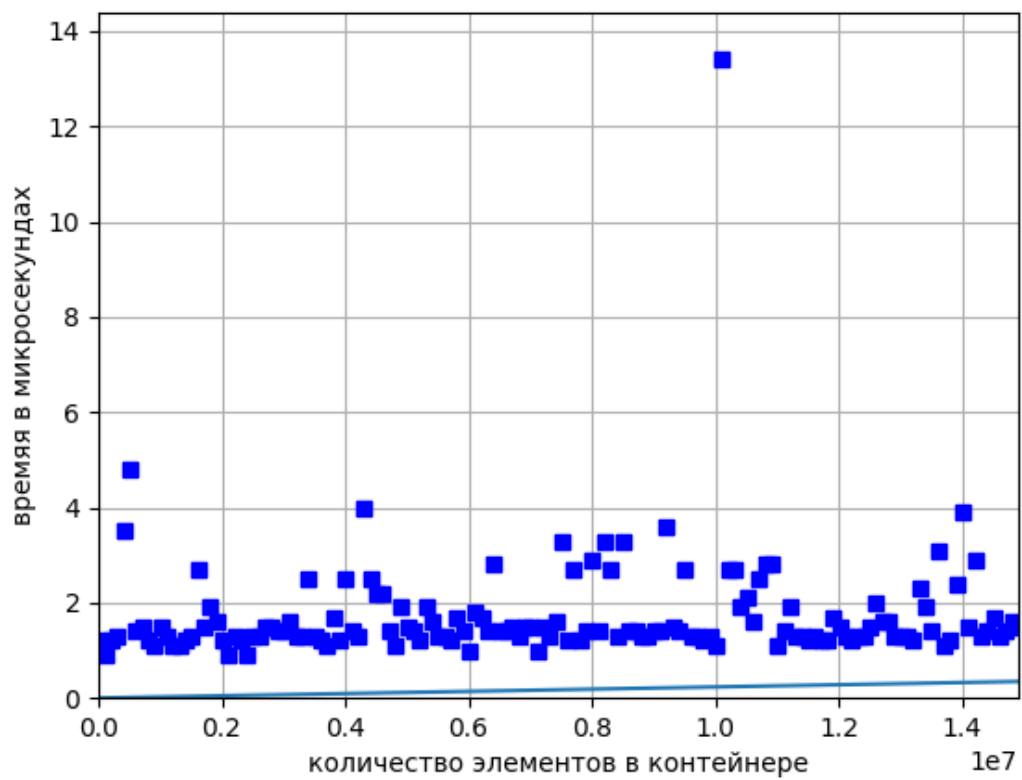
Вектор с максимальным количеством элементов = 1000 (push_back())



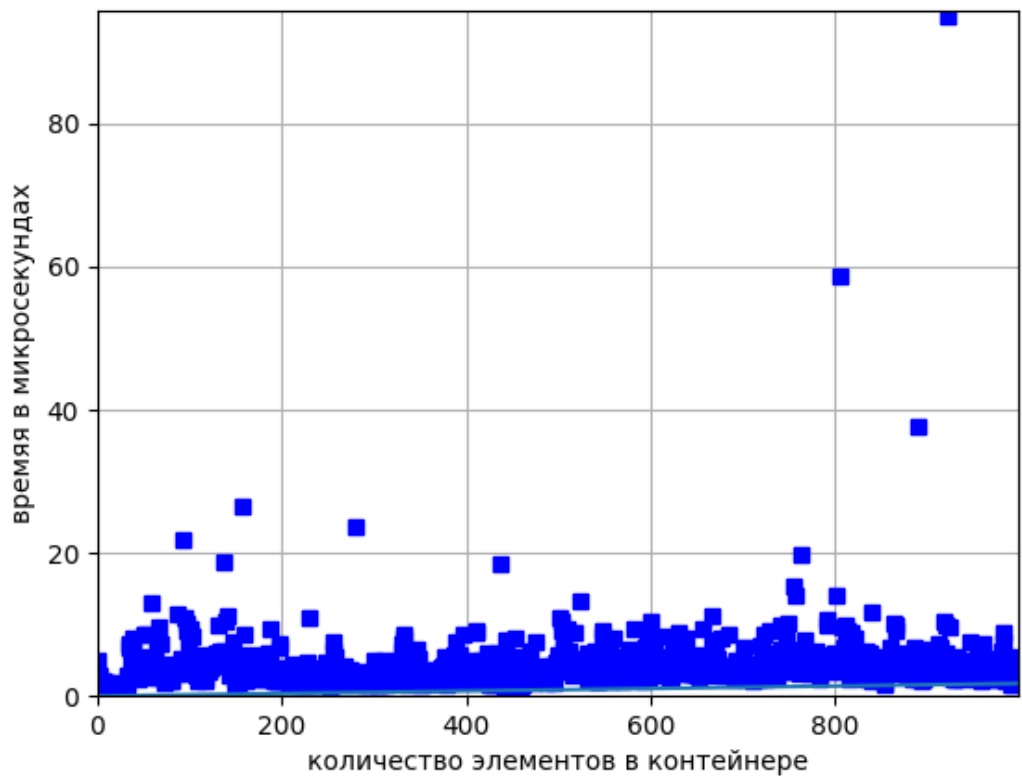
Вектор с максимальным количеством элементов = 3.3×10^7 (pop())



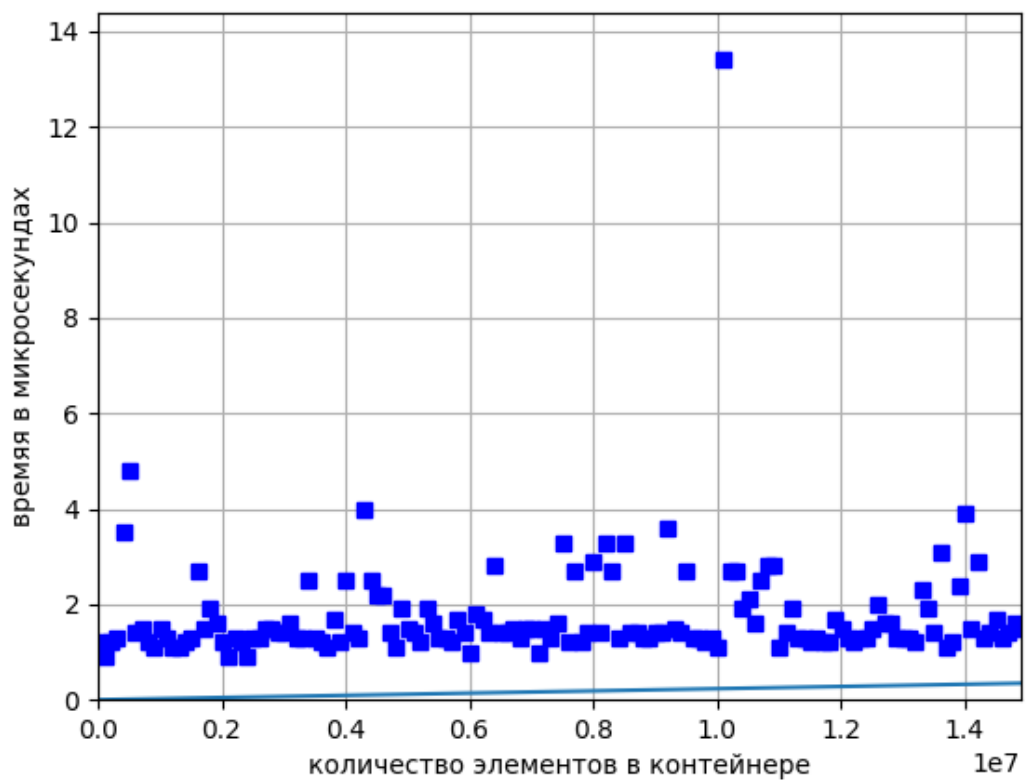
Вектор с максимальным количеством элементов = 1000 (pop())



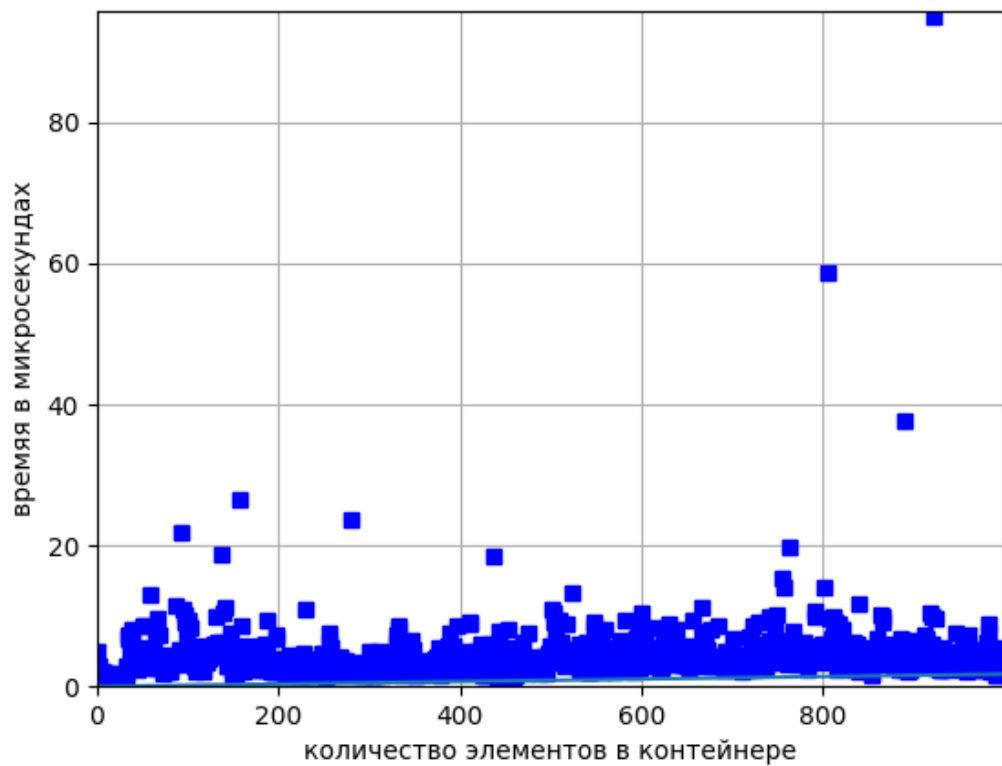
Список с максимальным количеством элементов = $1.8 \cdot 10^7$ (`push_back()`)



Список с максимальным количеством элементов = 1000 (`push_back()`)



Список с максимальным количеством элементов = $1.8 * 10^7$ (pop())



Список с максимальным количеством элементов = 1000 (pop())

Итог:

В данном исследовании вектор оказался лучше во всем, среднее время добавления элемента у него меньше, чем у списка, удаление – еще быстрее. Так же на графике невозможно отобразить предельные числа для контейнера, так как питон требовал невозможные системные требования для построения графика, однако для ознакомления можно посмотреть в текстовые файлы.

Вектор смог вместить в себя 100 миллионов элементов (и это не предел) и занял при этом около 500 Мб, в то время как список смог только 36 миллионов (после этого мой компилятор стал ругаться на объем используемой памяти – 2 Гб, кстати стоит обратить на значения времени при достижении предела ~ 30 мкс). Отсюда можно сделать вывод, что вектор расходует примерно в 12 раз меньше памяти!

Почитав статьи гиков на хабре, я смог понять, зачем нужен список, если у нас имеется более быстрая структура, расходующая меньше памяти, да и еще с доступом по индексу – вектор не многозадачен, в отличие от списка.