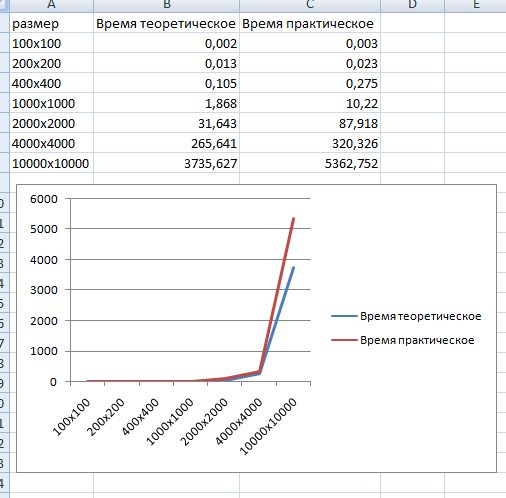
**Задание 1:**

1. Вычислил порядок сложности программы (*О*-символику).

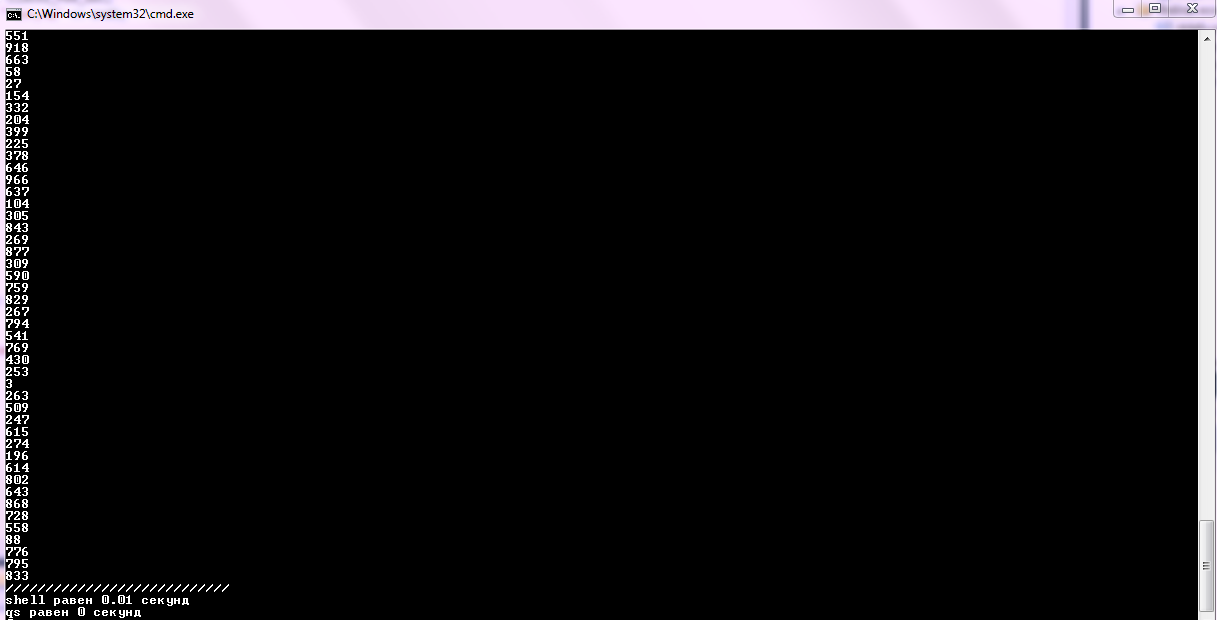
*О*-символика равна n3.

1. Оценил время выполнения программы и кода, выполняющего перемножение матриц, используя функции библиотеки time.h для матриц размерами от 100, 200, 400, 1000, 2000, 4000, 10000.И построил график исходя из теоретического времени затраченного на работу программы и практического .

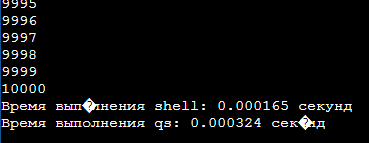


**Задание 2**:

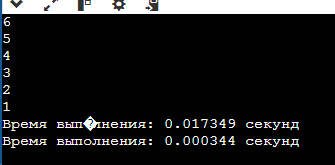
1. Оценил время работы каждого из реализованных алгоритмов на случайном наборе значений массива.



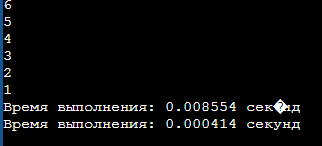
2. Оценил время работы каждого из реализованных алгоритмов на массиве, представляющем собой возрастающую последовательность чисел.



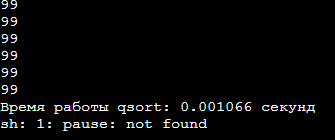
3. Оценил время работы каждого из реализованных алгоритмов на массиве, представляющем собой убывающую последовательность чисел.

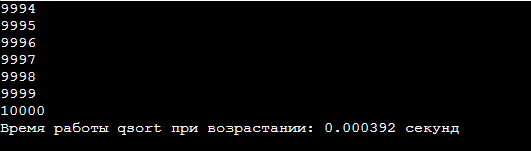


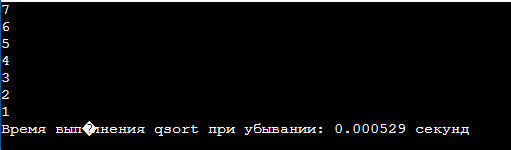
4. Оценил время работы каждого из реализованных алгоритмов на массиве, одна половина которого представляет собой возрастающую последовательность чисел, а вторая, – убывающую.

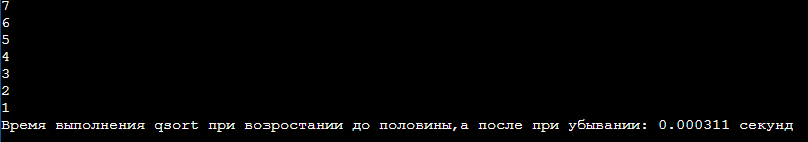


5. Оценил время работы стандартной функции qsort, реализующей алгоритм быстрой сортировки на выше указанных наборах данных.

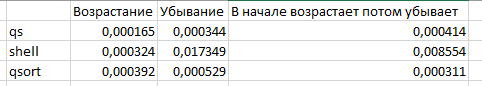








**Таблица сравнения работы массива.**



**Вывод:** В ходе работы были оценены время выполнения различных алгоритмов сортировки и вычислен их порядок сложности. Исследовано время перемножения матриц разных размеров (от 100 до 10,000 элементов), а также выполнена оценка функции qsort. Построен график зависимости времени выполнения от размера матриц и проведено сравнение с теоретическими оценками.