Министерство науки и высшего образования Российской Федерации  
Пензенский государственный университет  
Кафедра вычислительная техника

**ОТЧËТ**  
по лабораторной работе №3  
по дисциплине: «Динамические списки»

Выполнили студенты группы 22ВВВ2:  
Демин М. С.

Сергунов М. Р.

Амиров И. Р.

Приняли:  
Акифьев И. В.

Митрохин М. А.

Пенза 2023

**Название**

Динамические списки

**Цель работы**

Разобраться с основными моментами при работе с динамическими списками

**Лабораторное задание**

1.Реализовать приоритетную очередь, путём добавления элемента в список в соответствии с приоритетом объекта (т.е. объект  с большим приоритетом становится перед объектом с меньшим приоритетом).

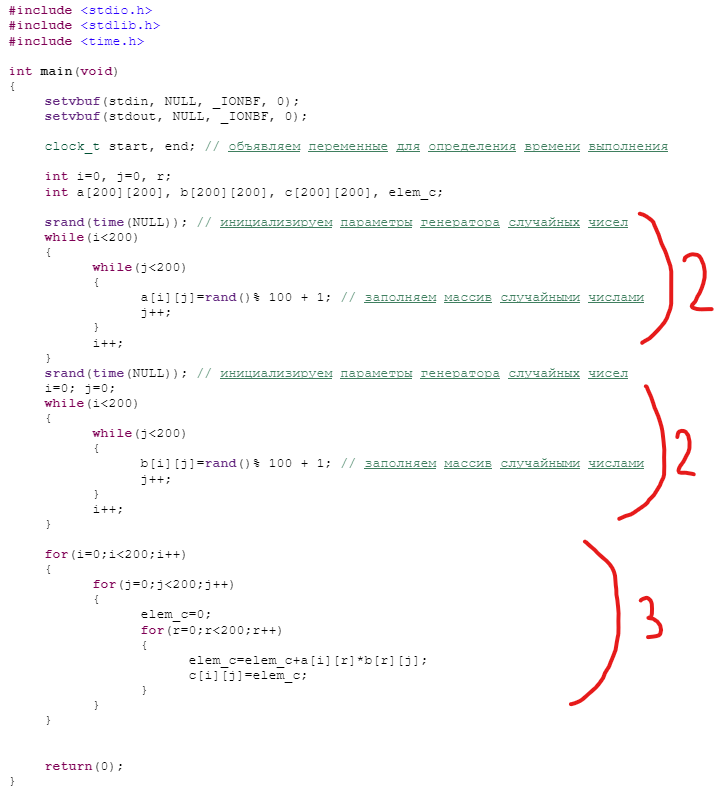
2.\* На основе приведенного кода реализуйте структуру данных Очередь.

3.\* На основе приведенного кода реализуйте структуру данных Стек.

**Листинг**

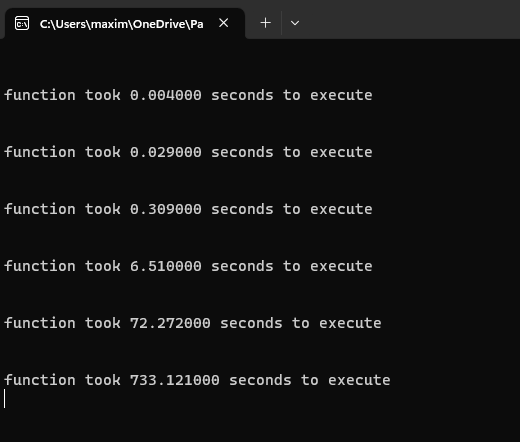
**Задание 1:**

Порядок сложности программы:

****

Порядок сложности программы равен n^3

Оценка время выполнения алгоритма для перемножения матриц:



100х100 – 0.004 сек

200х200 – 0.029 сек

400х400 – 0.309 сек

1000х1000 – 6.51 сек

2000х2000 – 72.272 сек

4000х4000 – 733.121 сек

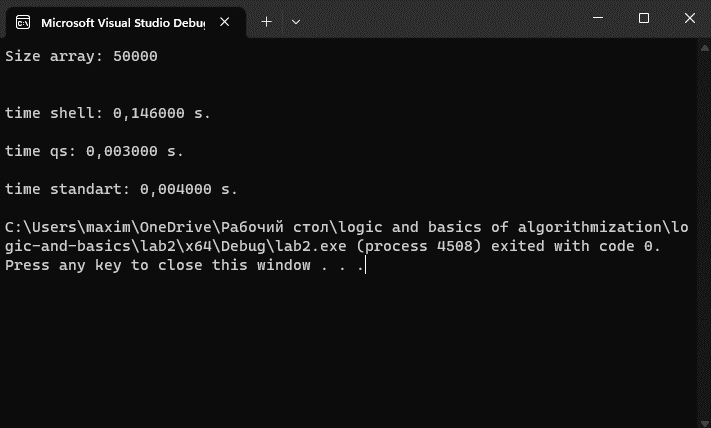
10000х10000 – н\д

График зависимости времени выполнения программы от размера матриц:

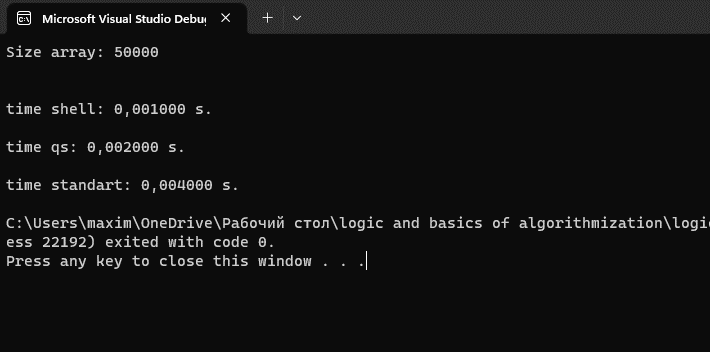
Графики совпали практически полностью, теоретические значения сложности(n^3) совпали с вычислениями, произведенными на практике, следовательно, решение правильно

**Задание 2:**

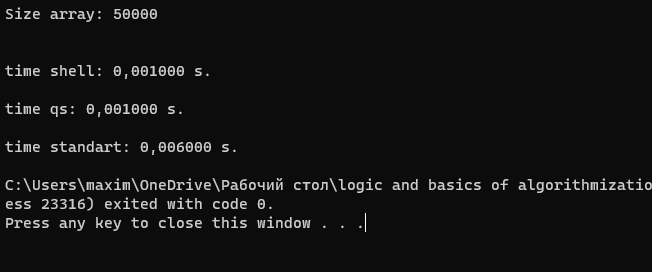
1. Оценить время работы каждого из реализованных алгоритмов на случайном наборе значений массива



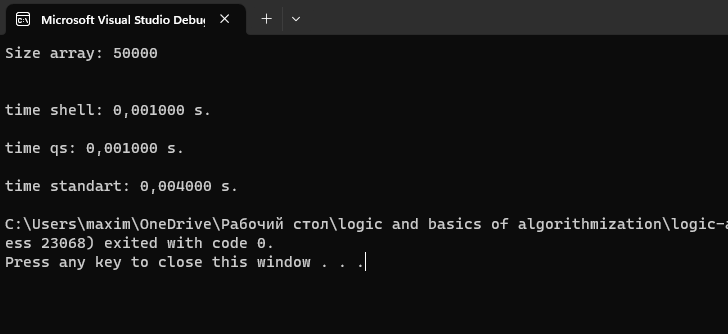
1. Оценить время работы каждого из реализованных алгоритмов на массиве, представляющем собой возрастающую последовательность чисел.



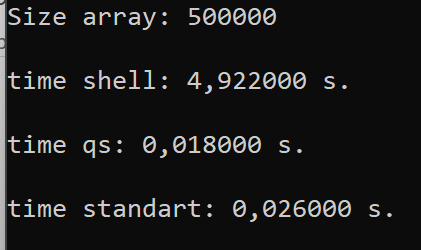
1. Оценить время работы каждого из реализованных алгоритмов на массиве, представляющем собой убывающую последовательность чисел.



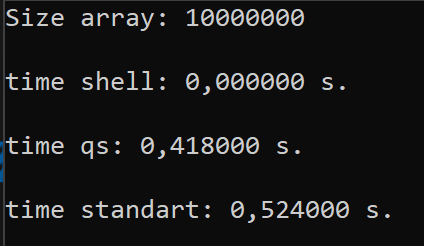
1. Оценить время работы каждого из реализованных алгоритмов на массиве, одна половина которого представляет собой возрастающую последовательность чисел, а вторая, – убывающую.



1. Поэкспериментировав с разными размерами массивов, мы можем наглядно видеть, что сортировка Шелла при большем объёме работает значительно дольше остальных.



Еще увеличили размер массива, отключив сортировку Шелла:



Видно, что сортировка qs и qsort работают с практически одинаковой скоростью, а Шелл работает лучше всего с возрастающим типом данных.

Оценили время работы стандартной функции qsort, реализующей алгоритм быстрой сортировки на выше указанных наборах данных (размер массива 50000).

на случайном наборе: 0.004 сек

возрастающая последовательность чисел: 0.004 сек

убывающая последовательность числе: 0.006 сек

половина возрастает половина убывает: 0.004 сек

**Вывод**

Вспомнили основные моменты при работе с простыми структурами данных.