Санкт-Петербургский Государственный Технический университет (Технологический институт)

Кафедра системного анализа и информационных технологий

**Лабораторная работа №3**

Выполнили:

Кнышев Станислав

Конаков Максим

422 группа

Проверил

Мусаев А.А

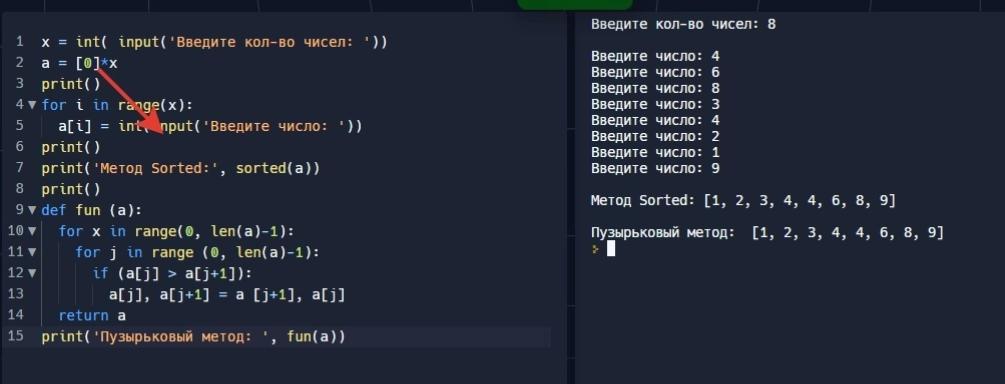
Санкт-Петербург

2022

**Задание 1**

**“Написать программу для пузырьковой сортировки. Оценить сложность данного метода. Сравнить с методом sort().”**

Пузырьковая сортировка:

Рис 1.

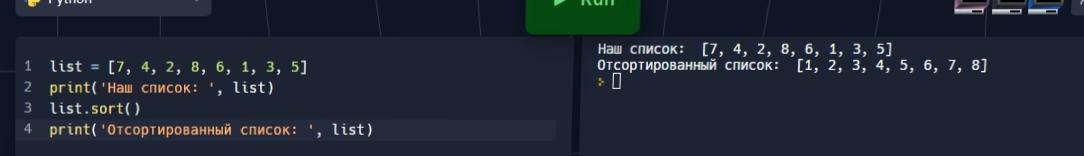
 Метод “sort()” также позволяет выполнить сортировку чисел в массиве.

Рис 2.

Перед нами две программы, которые выполняют сортировку массива, но выполняют они эту операцию разными способами. Сложность алгоритма сортировки методом “пузырька” можно охарактеризовать как О(n2). Метод sort имеет сложность O(nlog(n)).

**Задание 2**

**“Придумать и реализовать алгоритмы, имеющие сложность O(3n), O(n log n), O(n!), O(n^3), O(3log(n))”**

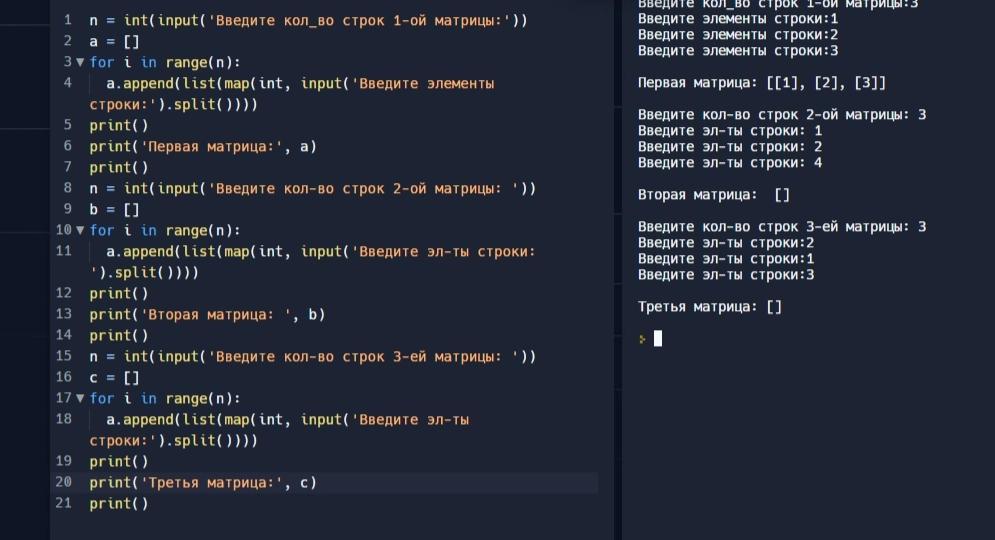
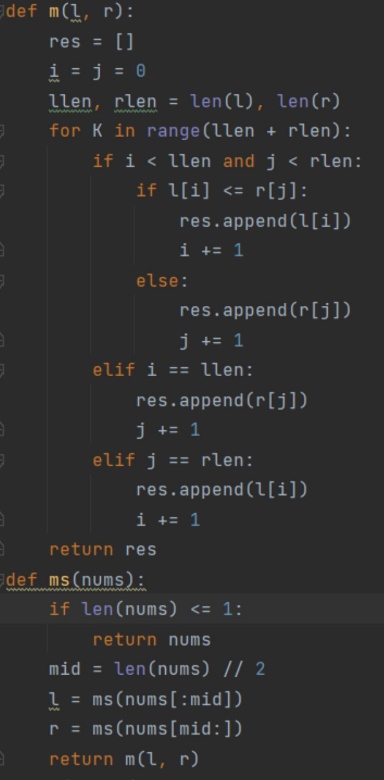
**** Написаны программы, имеющие сложность O(3n) (рис.3), O(n log n) (рис.4, 5), O(n!) (рис.6), O(n^3) (рис.7), O(3log(n)) (рис.8).

Рис 3.

**** Рис 4.

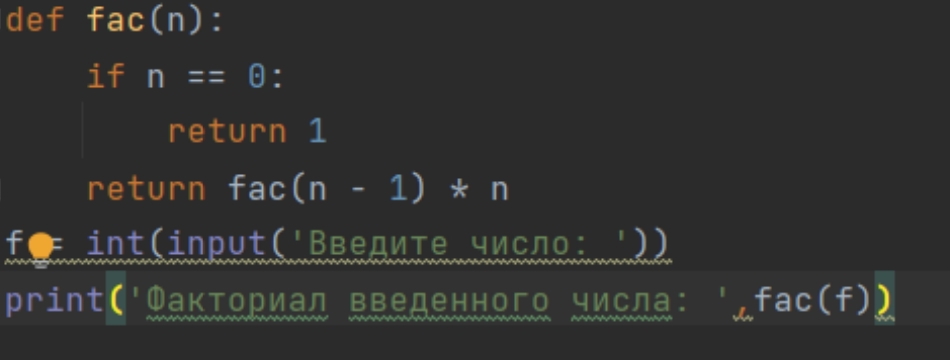
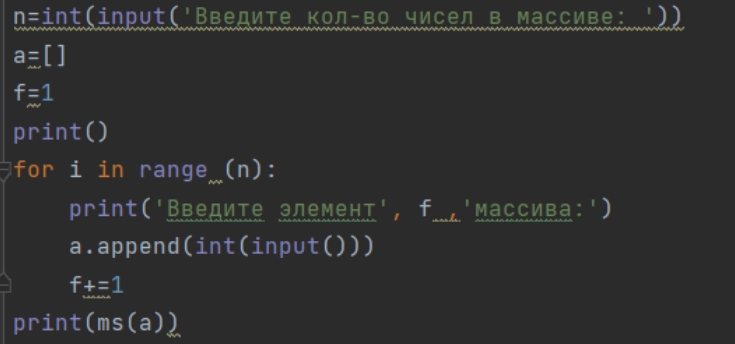
Рис 5.

Рис 6.

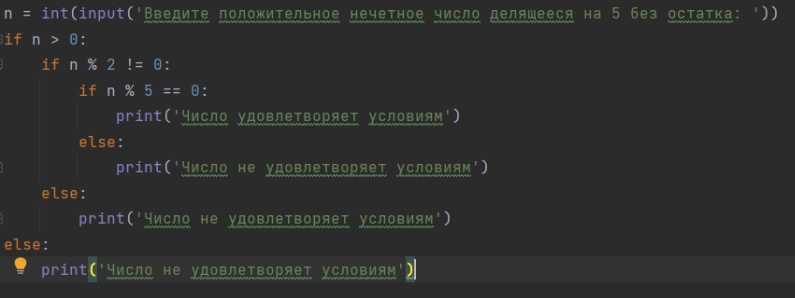
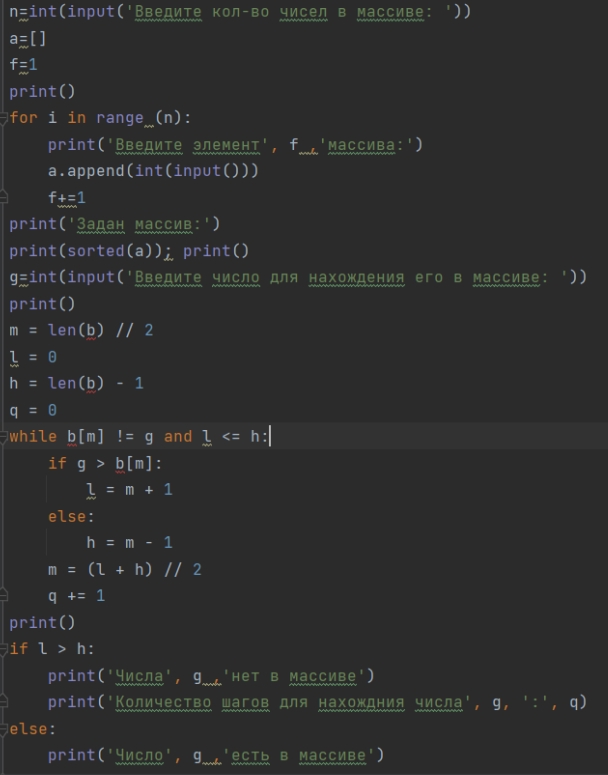
Программа вычисляет факториал для каждого числа в заданном массиве с помощью рекурсии.

Рис 7.

Программа проверяет цифру на удовлетворение заданным условиям.

Рис 8.

Программа трижды выполняет бинарный поиск.

**Задание 3**

**“Построить зависимость между количеством элементом и количеством шагов для алгоритмов со сложностью O(1), O(log n), O(n^2), O(2^n). Сравнить сложность данных алгоритмов.”**

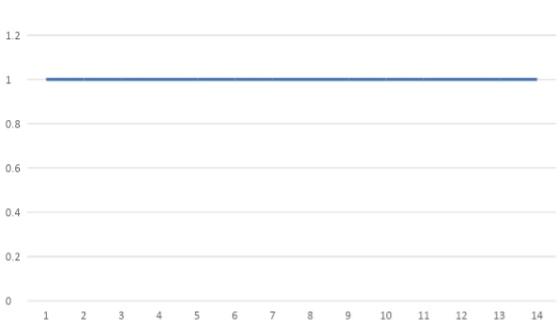
**** Для наглядного понимания зависимости между количеством элементов и количеством шагов для алгоритмов со сложностью O(1) (рис.9), O(logn)(рис.10), O(n^2) (рис.11), O(2^n) (рис.12) построены графики.

Рис 9.

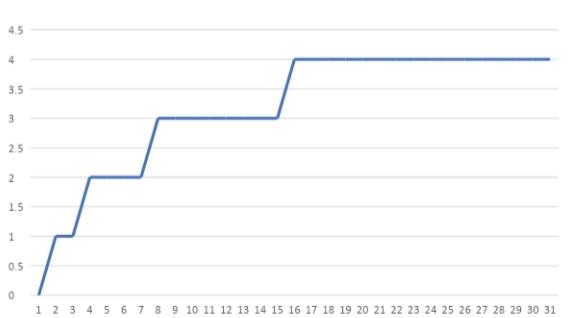
 В сложности O(1) количество шагов не зависит от количества переменных, это означает то, что при изменении количества переменных время выполнения программы никак не изменится.

Рис 10.

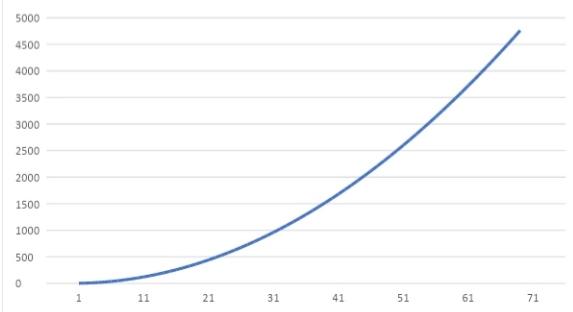
 В сложности O(logn) происходит отрицательное ускорение функции, следовательно данная сложность выполнения алгоритма самая оптимизированная, особенно, когда речь идет о большом количестве n.

Рис 11.

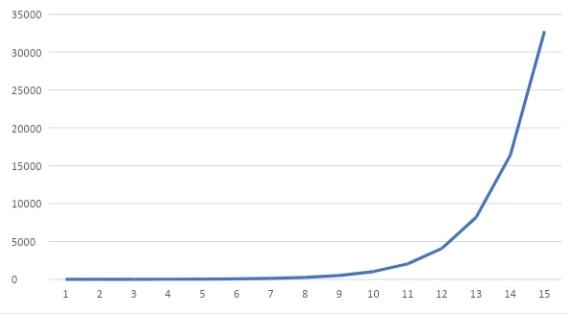
 В сложности O(n^2) количество шагов увеличивается полиномиально.

Рис 12.

В сложности O(2^n) количество шагов увеличивается экспоненциально. **Сравнение**: Отталкиваясь от оценки всех вышеперечисленных сложностей, можно сказать, что худшей сложностью из приведенных является O(2^n), а лучшей сложностью O(log n).