Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики

Мегафакультет ИКТ

**Лабораторная работа №1**

**Вариант №1**

Выполнил(и:) Затикян С. А.

Кабдулвахитов Э. И

Комелин Г. А.

Проверил Мусаев А.А.

Санкт-Петербург,

2022

**СОДЕРЖАНИЕ**

[**Введение 3**](#_Toc115419980)

[**1. Задание 1 4**](#_Toc115419981)

[**1.1. Функция транспонирования матрицы 4**](#_Toc115419982)

[**1.2. Функция умножения матриц 5**](#_Toc115419983)

[**1.3. Функция определения ранга матрицы 6**](#_Toc115419984)

[**2. Задание 2 7**](#_Toc115419985)

[**2.1. Функция транспонирования матрицы с использованием библиотеки numpy 7**](#_Toc115419986)

[**2.2. Функция умножения матриц с использованием библиотеки numpy 8**](#_Toc115419987)

[**2.3. Функция определения ранга матрицы с использованием библиотеки numpy 9**](#_Toc115419988)

[**2.4. Достоинства и недостатки numpy 10**](#_Toc115419989)

[**3. Задание 3 11**](#_Toc115419990)

[**3.1. Функция возведения матрицы 3x3 в степень -1 11**](#_Toc115419991)

[**3.2. Сравнение скорости работы моей функции с ее аналогом из библиотеки numpy с помощью библиотеки timeit 12**](#_Toc115419992)

[**Заключение 13**](#_Toc115419993)

[**Список литературы 14**](#_Toc115419994)

[**Приложение 15**](#_Toc115419995)

Введение

Целью данной работы является ознакомление с языком программирования Python. Для выполнения лабораторной работы были поставлены следующие задачи:

1. Написать программу для пузырьковой сортировки. Оценить сложность данного  
   метода. Сравнить с методом sort().
2. Придумать и реализовать алгоритмы, имеющие сложность O(3n), O(nlogn), O(n!), O(n3), O(3log(n))
3. Построить зависимость между количеством элементом и количеством шагов для алгоритмов со сложностью О(1), O(logn), O(n^2), O(2^n). Сравнить сложность данных алгоритмов.
4. Задание 1
   1. Пузырьковая сортировка

Для того чтобы реализовать данный алгоритм, я использовал три цикла For, один из которых вводил случайный числа в массив с помощью randint, а два других уже сортировали его. Программа сортирует массив, постепенно меняя соседние числа местами. Моя программа представлена на рисунке 1.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 – Пузырьковая сортировка

В результате выполнения данной функции выводится отсортированный массив.

1. Задание 2

2.1. Программы разной сложности

Внизу представлены программы включающие в себя функции разных сложностей. Рисунок 2

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 – Программа сложностью O(3n), O(n!), O(n^3)

На рисунке 4 показаны программы 3 сложностей:

* O(3n) – вывод строки некоторое количество раз
* O(n!) – оператор sort
* O(n^3) – три цикла внутри друг друга

На рисунке 3 показаны функция с сложностью O(logn), но в первом случае она повторяется n разz, то есть сложность O(nlogn), а во втором всего лишь 3 раза и его сложность O(3logn)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 – Программа сложностью O(nlogn), O(3logn)

3. Задание 3

* 1. Графики функций разных сложностей

Всего надо построить 4 графиков функций О(1), O(logn), O(n^2), O(2^n). Самая легкий из этих алгоритмов это первый O(1), далее идет O(logn), O(n^2), O(2^n). Данные графики представлены на рисунках 4, 5, 6, 7.

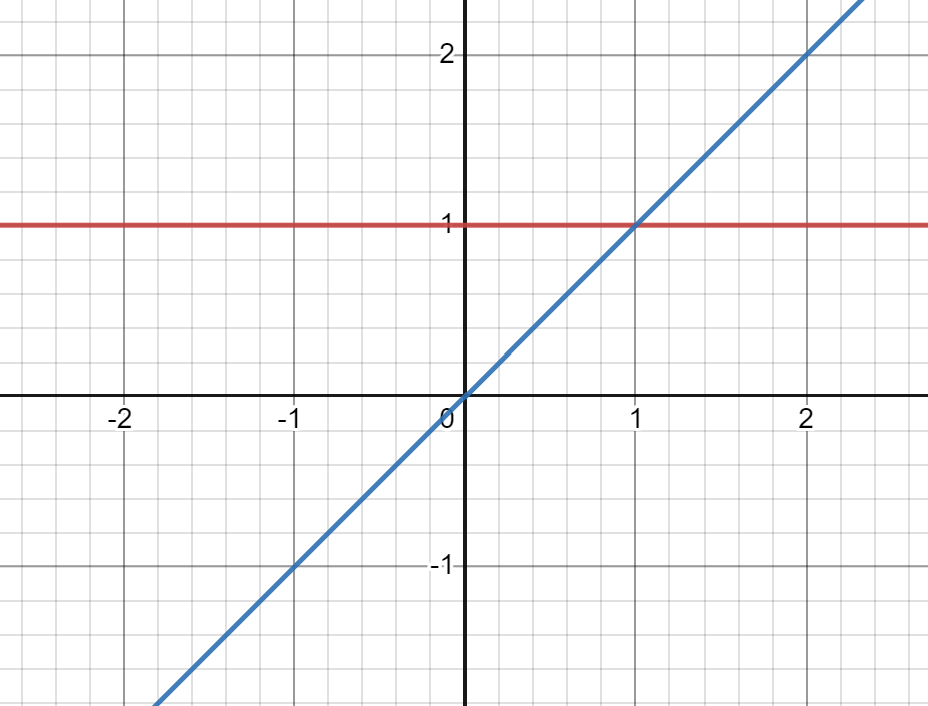


Рисунок 4 – График сложностью O(1)

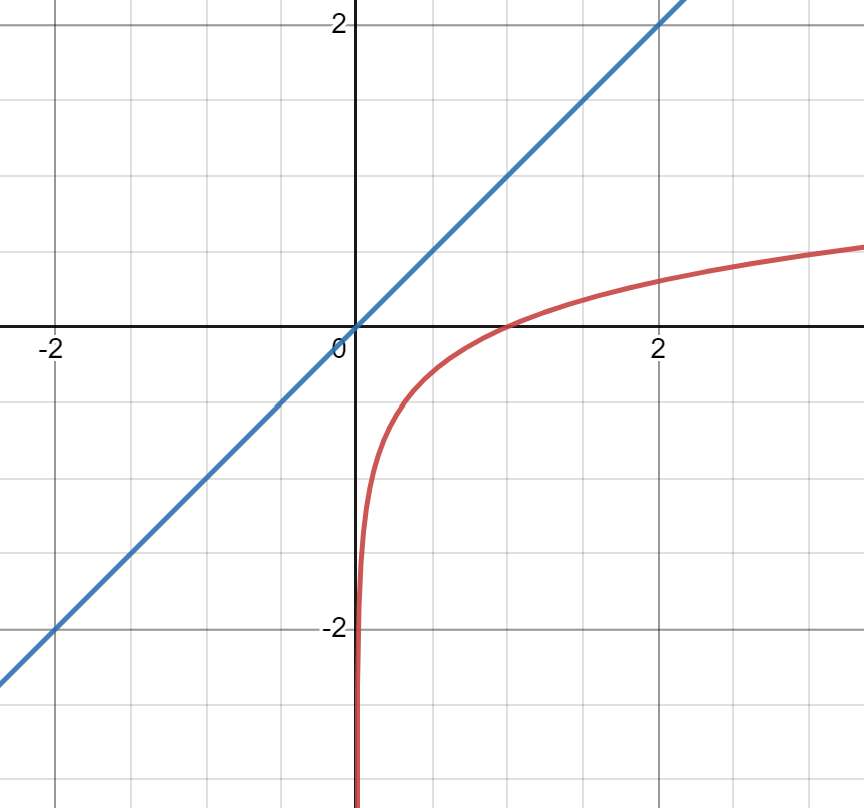


Рисунок 5 – График сложностью O(logn)

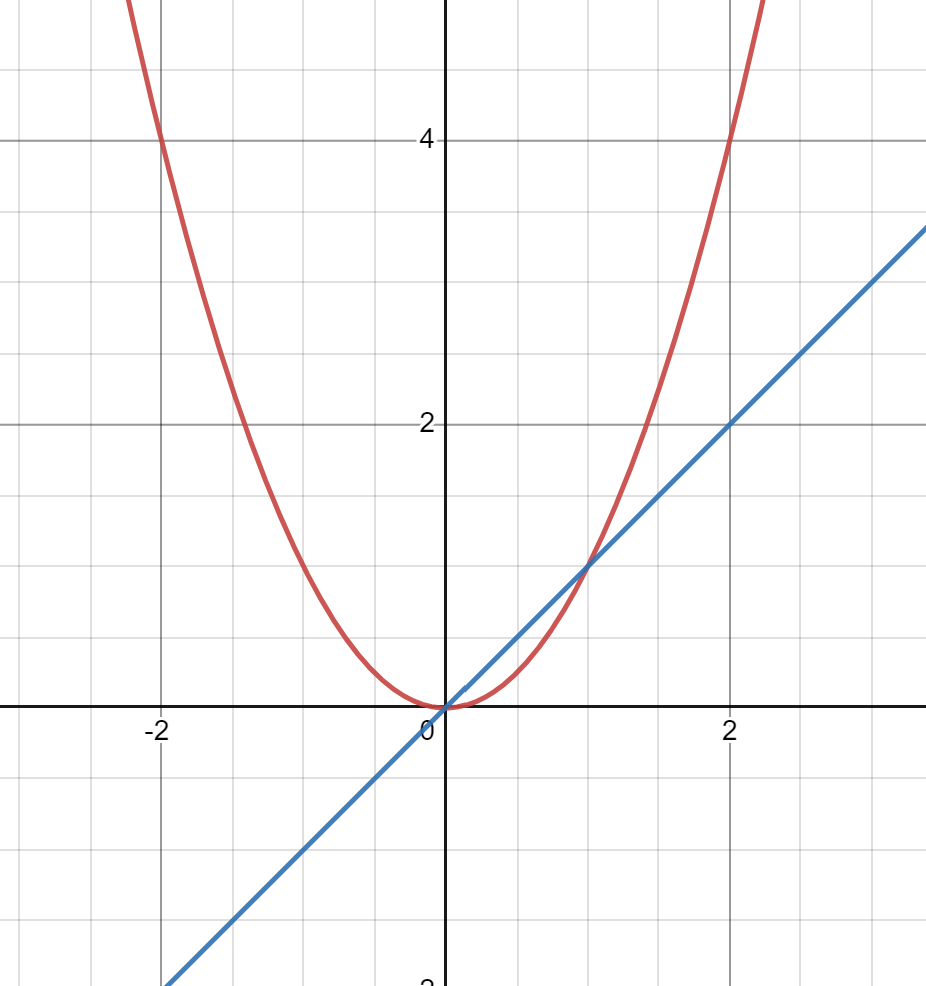


Рисунок 6 – График сложностью O(n^2)

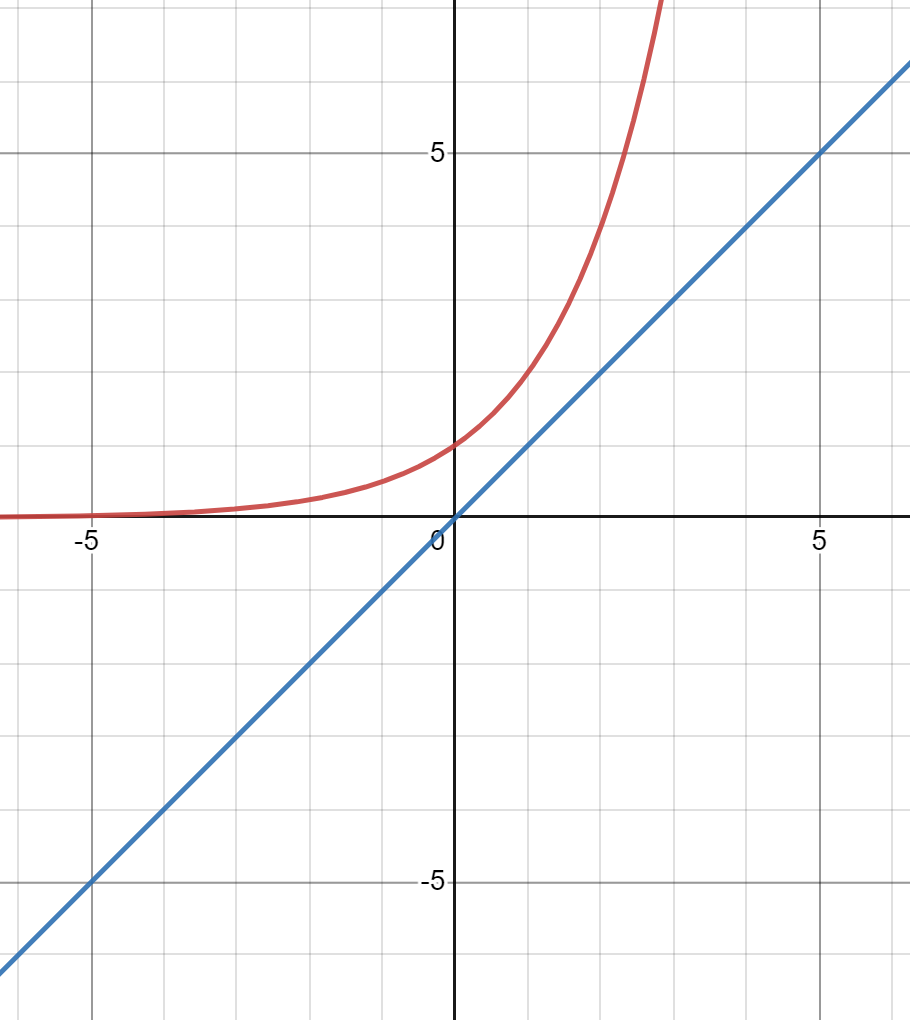


Рисунок 7 – График сложностью O(2^n)

,

Заключение

В результате выполнения данной лабораторной работы были выполнены все задания. Были написаны программы разных сложностей и проведен анализ зависимостью между переменной и функцией с разными сложностями.

Список литературы

1. matematikam // URL: <https://matematikam.ru/calculate-online/grafik.php>