

Санкт-Петербургский Национальный Исследовательский Университет
Информационных технологий, механики и оптики

Факультет инфокоммуникационных технологий

Лабораторная работа №4
Вариант №1

Выполнили:

Смирнов И.И.

Касьяненко В.М.

Проверил:

Мусаев А.А.

Санкт-Петербург

2022

СОДЕРЖАНИЕ

Введение.....	3
1 Задание 1	4
1.1 Быстрая сортировка и сортировка расческой.....	4
Задание 2	6
2.1 Блочная и пирамидальная сортировки.....	6
Задание 3	7
3.1 Анализ сортировок	7
Заключение	8
Список литературы	9
Приложение	10

ВВЕДЕНИЕ

Для становления хорошим специалистом в области программирования на языке Python необходимо знать основные алгоритмы и функционал языка.

Цель данной работы – ознакомление с различными сортировками на языке программирования Python.

В ходе выполнения лабораторной работы были решены следующие задачи:

- создание программ с сортировкой;
- оценивание достоинства и недостатки различных сортировок.

Задания, которые необходимо выполнить:

1. Задание 1: написать программу с функциями для быстрой сортировки и сортировки расческой. Использовать данные функции в программе как модуль в другой программе. Пользователь выбирает один из двух методов сортировки. Оценить время выполнения программы с помощью модуля `timeit`.

2. Задание 2: изучить блочную и пирамидальную сортировку. Написать соответствующие программы.

3. Задание 3: оценить достоинства, недостатки и сложность изученных методов сортировок.

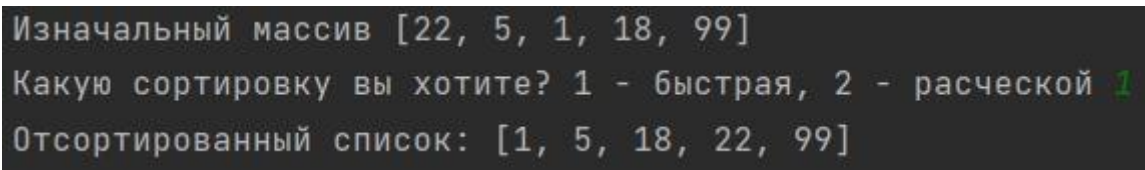
1 ЗАДАНИЕ 1

1.1 Быстрая сортировка и сортировка расческой

Быстрая сортировка начинается с разбиения списка и выбора одного из элементов в качестве опорного. А всё остальное передвигается так, чтобы этот элемент встал на своё место. Все элементы меньше него перемещаются влево, а равные и большие элементы перемещаются вправо.

При сортировке расческой производятся неоднократные прогоны по массиву, при которых сравниваются пары элементов. Если они не отсортированы друг относительно друга - то производится обмен. В результате крупные элементы мигрируют в конец массива, а небольшие по значению - в начало.

Была создана программа, в которой данные сортировки используются как модули, а пользователь выбирает какой сортировкой он желает отсортировать список элементов (рисунок 1).



```
Изначальный массив [22, 5, 1, 18, 99]  
Какую сортировку вы хотите? 1 - быстрая, 2 - расческой 1  
Отсортированный список: [1, 5, 18, 22, 99]
```

Рисунок 1 – Программа с выбором сортировок

1.2 Оценка времени выполнения программы

Для сравнения быстродействия быстрой сортировки и сортировки расческой будет использована библиотека Timeit. С помощью тестов были определены некоторые результаты, записанные в таблицу 1.

Быстрая сортировка	Сортировка расческой
0.02453110000351444	0.020050899998750538
0.024245800042990595	0.017545500013511628
0.023677099961787462	0.01593170000705868
0.026660299976356328	0.02272639999864623
0.021814099978655577	0.016342399991117418

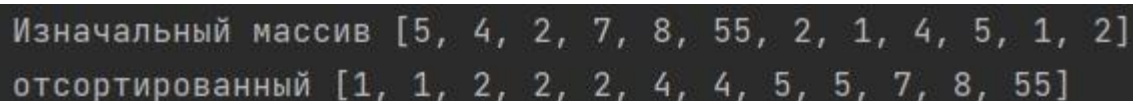
Таблица 1 – Оценка времени выполнения программы

Посчитав среднее значение, можно увидеть, что быстрая сортировка работает медленнее в 1,3 раза.

ЗАДАНИЕ 2

2.1 Блочная и пирамидальная сортировки

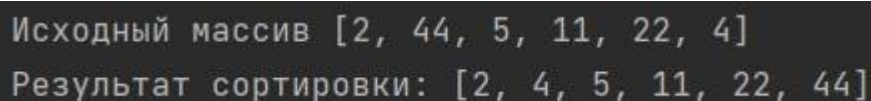
Алгоритм блочной сортировки использует значения элементов для разбиения их на блоки, и затем рекурсивно сортирует полученные блоки. Когда блоки становятся достаточно малыми, алгоритм останавливается и использует более простой алгоритм типа сортировки выбором для завершения процесса (рисунок 2).



```
Изначальный массив [5, 4, 2, 7, 8, 55, 2, 1, 4, 5, 1, 2]  
отсортированный [1, 1, 2, 2, 2, 4, 4, 5, 5, 7, 8, 55]
```

Рисунок 2 – Блочная сортировка

При пирамидальной сортировке на первом этапе из входных данных строится отдельный список. На втором этапе создается отсортированный массив путем многократного удаления самого большого или наименьшего элемента из отдельного списка и вставки его в массив. Как только все элементы будут удалены из отдельного списка, получается отсортированный массив (рисунок 3).



```
Исходный массив [2, 44, 5, 11, 22, 4]  
Результат сортировки: [2, 4, 5, 11, 22, 44]
```

Рисунок 3 – Пирамидальная сортировка

ЗАДАНИЕ 3

3.1 Анализ сортировок

	Достоинства	Недостатки	Сложность
Быстрая сортировка	Алгоритм относительно короткий Не требует много памяти	Неустойчива Скорость может сильно упасть при неудачных входных данных	$O(n \log n)$
Сортировка расческой	Простой код Эффективная Достаточно быстрая	Неустойчива	$O(n \log n)$
Блочная сортировка	Быстрая Работает с типами любых данных	Скорость может упасть при неудачных входных данных	$O(n \log n)$
Пирамидальная сортировка	Быстро сортирует большие списки Не требует много памяти	Неустойчива	$O(n \log n)$

На основе этой таблицы мы можем сделать вывод, что сортировки имеют свои преимущества и недостатки, а также имеют наиболее подходящие для них случаи применения.

ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы был получен опыт написания и оценивания программ с различными сортировками.

СПИСОК ЛИТЕРАТУРЫ

1. Habr. Пирамидальная сортировка. [Электронный ресурс] – <https://habr.com/ru/company/otus/blog/460087/> (Дата последнего обращения 24.10.2022).

ПРИЛОЖЕНИЕ

Ссылка на полный код данной лабораторной работы:

[https://github.com/VeraKasianenko/ITMO_ICT/tree/main/1_term_ICT/
Algoritms/labs/lab_4](https://github.com/VeraKasianenko/ITMO_ICT/tree/main/1_term_ICT/Algoritms/labs/lab_4)