

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

Факультет инфокоммуникационных технологий

Лабораторная работа №4
Вариант №1

Выполнили:

Кремпольская Е.А

Петрова Н.Г

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
СПИСОК ЛИТЕРАТУРЫ.....	8
ПРИЛОЖЕНИЕ	9

ВВЕДЕНИЕ

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 разбивает массив на меньшие массивы до тех пор, пока он не закончится пустым массивом, или массивом, содержащим только один элемент, и затем все рекурсивно соединяется в отсортированный большой массив.

При пузырьковом алгоритме сравниваются постоянно два элемента. В сортировке расческой эти элементы берутся не соседними, а как бы по краям «расчески» — первый и последний. Расстояние между сравниваемыми элементами наибольшее из возможных, то есть, это максимальный размер расчески.

Была создана программа, в которой данные сортировки используются как модули, а пользователь выбирает какой сортировкой он желает отсортировать список элементов. Метод быстрой сортировки (рисунок 1). Метод сортировки расчёской (рисунок 2).

```
Выберете сортировку:
1) Быстрая сортировка
2) Сортировка расческой

Введите номер действия: >? 1
Введите количество >? 4
Введите число >? 1
Введите число >? 1
Введите число >? 5
Введите число >? 9
Сортировка завершена!
1.0 1.0 5.0 9.0
Время выполнения quicksort
3.3334014005959034e-05
```

Рисунок 1 – быстрая сортировка

```
Выберете сортировку:
1) Быстрая сортировка
2) Сортировка расческой

Введите номер действия: >? 2
Введите количество >? 4
Введите число >? 1
Введите число >? 1
Введите число >? 5
Введите число >? 9
Сортировка завершена!
1.0 1.0 5.0 9.0
Время выполнения comb_sort
1.1708005331456661e-05
```

Рисунок 2 – сортировкой расчёской

Для сравнения быстродействия быстрой сортировки и сортировки расческой будет использована библиотека Timeit. Посчитав среднее значение, мы заметили, что быстрая сортировка работает медленнее в 1,3 раза.

Вывод: Сортировка расчёской – это очень эффективный метод сортировки. Стабильная сортировка, требуется меньше памяти и времени, основана на сравнении. Написали программы быстрая сортировка и сортировка расчёской, оценив время данных методов при помощи модуля Timeit.

ЗАДАНИЕ 2

2.1 Блочная и пирамидальная сортировки

В алгоритме блочная сортировки сортируемые элементы распределяются между конечным числом отдельных блоков (карманов, корзин) так, чтобы все элементы в каждом следующем по порядку блоке были всегда больше (или меньше), чем в предыдущем. Каждый блок затем сортируется отдельно, либо рекурсивно тем же методом, либо другим. Затем элементы помещаются обратно в массив. Этот тип сортировки может обладать линейным временем исполнения. (рисунок 3).

```
Введите количество >? 4
Введите число >? 25
Введите число >? 4678
Введите число >? 1
Введите число >? 0.2
Сортировка завершена!
[0.2, 1.0, 25.0, 4678.0]
```

Рисунок 3 – Блочная сортировка

Алгоритм пирамидальной сортировки полагается на такие структуры данных как двоичные кучи. Поскольку мы знаем, что кучи всегда должны соответствовать определенным требованиям, мы можем использовать это для поиска элемента с наименьшим значением, последовательно сортируя элементы, выбирая корневой узел кучи и добавляя его в конец массива (рисунок 4).

```
Введите количество >? 4
Введите число >? 5
Введите число >? 44
Введите число >? 13
Введите число >? 2
Сортировка завершена!
2.0 5.0 13.0 44.0
```

Рисунок 4 – Пирамидальная сортировка

ЗАДАНИЕ 3

3.1 Анализ сортировок

Быстрая сортировка ($O(n \log n)$) – алгоритм относительно короткий, не требует много памяти, но неустойчив, скорость может сильно упасть при неудачных входных данных;

Сортировка расчёской ($O(n \log n)$) – простой код, эффективная сортировка, достаточно быстрая, но неустойчивая;

Блочная сортировка ($O(n \log n)$) – быстрая, работает с типами любых данных, но скорость может упасть при неудачных входных данных;

Пирамидальная сортировка ($O(n \log n)$) – быстро сортирует большие списки, не требует много памяти, но неустойчива.

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

СПИСОК ЛИТЕРАТУРЫ

1. Habr. [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://clck.ru/32pBt6>