Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №10**

**дисциплины «Алгоритмизация»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Кожуховский Виктор Андреевич  2 курс, группа ИВТ-б-о-22-1,  09.03.01 «Информатика и вычислительная техника», направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем  », очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Руководитель практики:  Воронкин Роман Александрович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2023 г.

Порядок выполнения работы:

Написал программу поиска элемента в массиве, автоматического заполнения массива, расчёта тысячи точек, показывающих время поиска элемента в массиве в худшем и среднем случае, вывода графиков, составленных из этих точек, и подсчета корреляции:

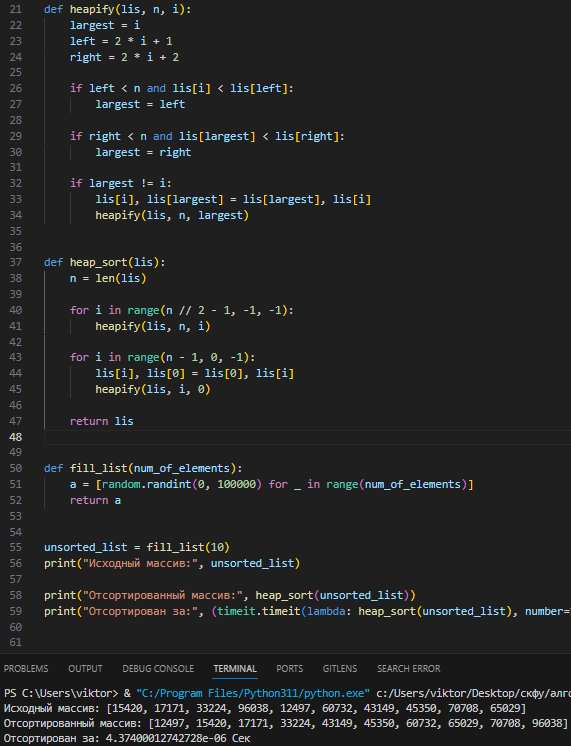


Рисунок 1. Код неоптимизированного алгоритма heapsort

Таблица 1. Сравнение алгоритма Heap Sort с Quick Sort и Merge Sort

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристика | Heap Sort | Quick Sort | Merge Sort |
| Сложность времени | O(n log n) | O(n^2) в худшем случае, O(n log n) в среднем | O(n log n) |
| Сложность по памяти | O(1) | O(log n) в среднем | O(n) |
| Стабильность | Нестабильная | Нестабильная | Стабильная |
| Необходимость доп. памяти | Нет | O(log n) в среднем, но может быть O(n) | Нет |
| Лучший случай | O(n log n) | O(n log n) | O(n log n) |
| Худший случай | O(n log n) | O(n^2) | O(n log n) |
| Средний случай | O(n log n) | O(n log n) | O(n log n) |

Heapsort не требует доп. память, занимает меньше всех места, но считается нестабильным. Quick Sort медленный в худшем случае, занимает больше места, требует доп. память и является нестабильным. Merge Sort стабилен, быстр, не требует доп памяти, но имеет наибольшую сложность по памяти.

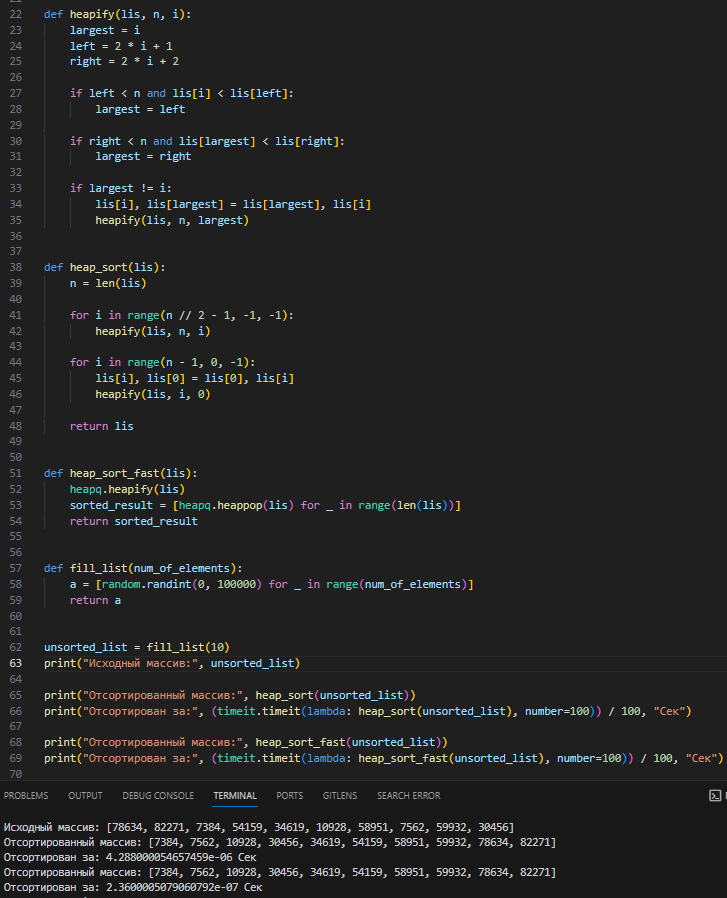


Рисунок 2. Оптимизированный алгоритм heapsort

Вывод: в результате выполнения лабораторной работы был изучен алгоритм линейного поиска и проведено исследование зависимости времени поиска от количества элементов в массиве, показавшее что зависимость время поиска линейно увеличивается с добавлением элементов в массив.