ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 5

«Реализация алгоритмов odd-even sort, heap sort, timsort»

Выполнил работу

Степанов Виктор

Академическая группа №группы

Принято

, Вершинин Владислав Константинович

Санкт-Петербург

2024

**Структура отчёта:**

1. Введение

Цель: понять работу алгоритмов.

Задача: понять какой алгоритм и где эффективен.

1. Теоретическая подготовка

В работе используются vector и тип данных int. В алгоритме Timsort содержится алгоритм сортировки вставками.

1. Реализация

В работе представлена реализация алгоритмов odd-Evensort, Heapsort, Timsort. Подключены библиотеки vector для массивов и cassert для unittest. Алгоритмы выполнены в виде функций представленных в программе.

1. Экспериментальная часть

Odd-Evensort:

Память: 4(переменная) + 4\*n(размер массива) + 8(две локальных переменных) байт.

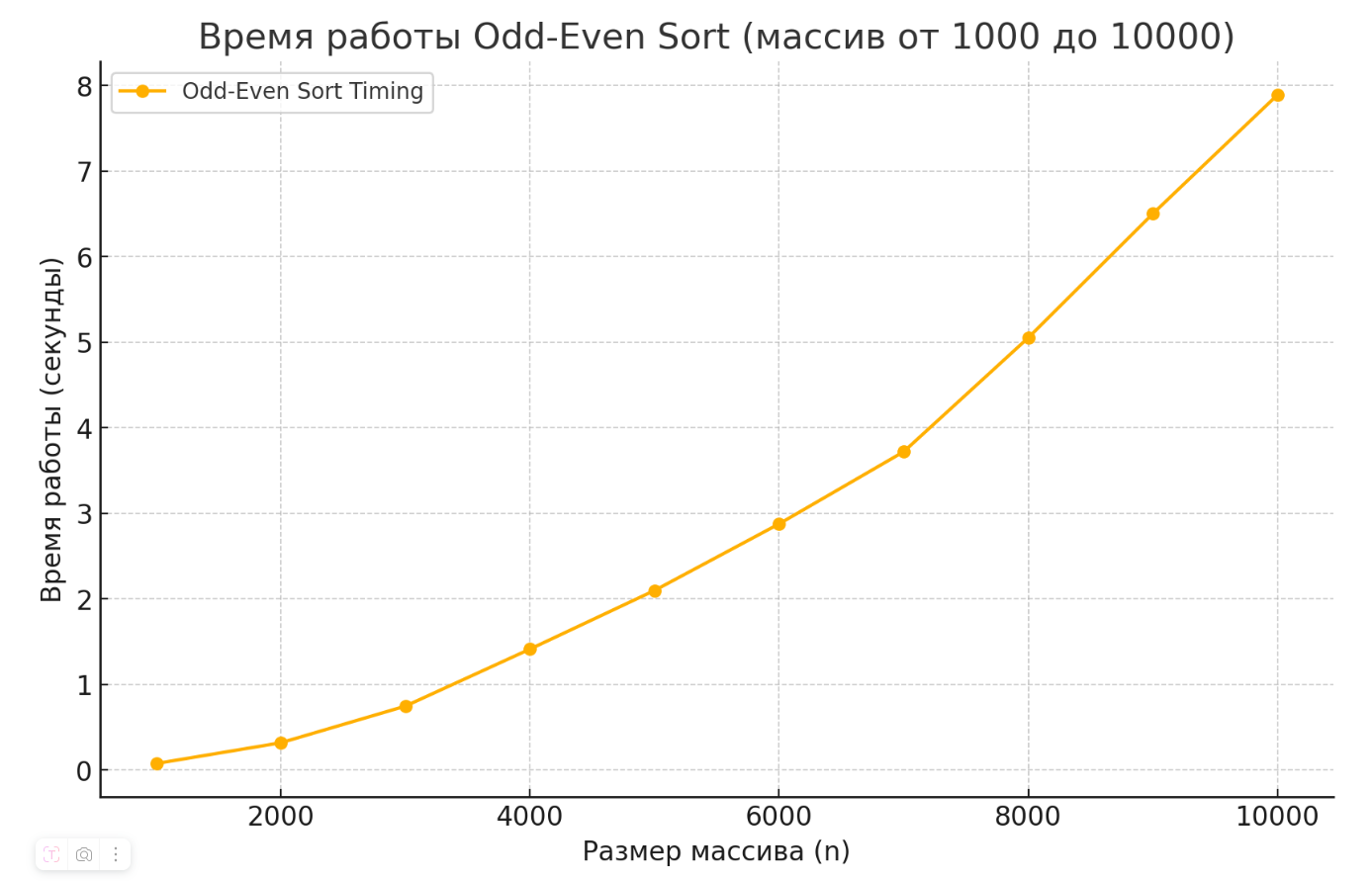
Асимптотика:O(n\*n)=O(n\*\*2)

Согласно требованиям моего варианта, на вход к моему алгоритму подаётся до 1е6 элементов. Теоретически заданная сложность задачи составляет O(n^2). Для тестирования алгоритма была собрана статистика, приведенная в таблице №1.

Таблица №1 - Подсчёт сложности реализованного алгоритма

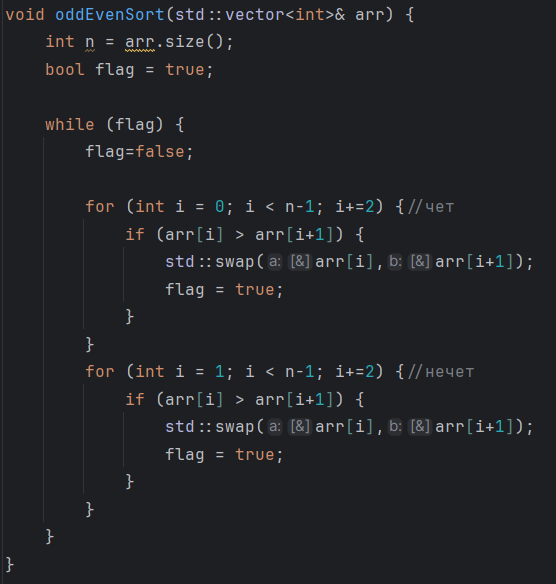
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Размер входного набора | 1000 | 2000 | 4000 | 6000 | 8000 | 10000 |
| Время выполнения программы, с | 0,1 | 0,3 | 1,5 | 3 | 5 | 8 |

График представляющий визуально удобный формат данных из таблицы №1 представлен на изображении №1.



Изображение №1 - График работы алгоритма

Сложность этого алгоритма делает его неэффективным на больших массивах.



**Heapsort:**

Память: 4\*n + 16 \* log2(n) + 16 байт

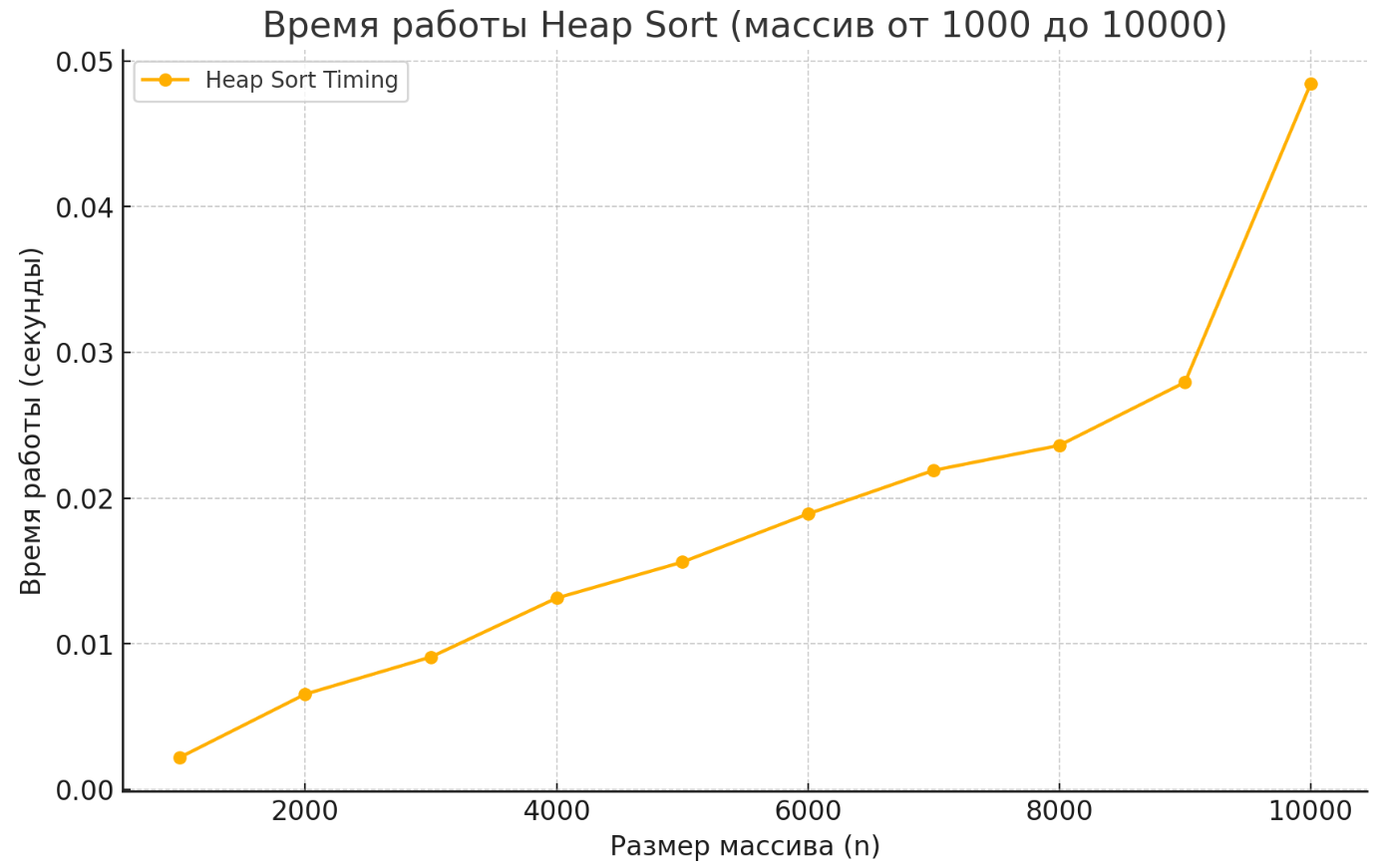
Асимптотика: O(n)+O(nlogn)=O(nlogn)

Согласно требованиям моего варианта, на вход к моему алгоритму подаётся до 1е6 элементов. Теоретически заданная сложность задачи составляет O(n^2). Для тестирования алгоритма была собрана статистика, приведенная в таблице №2.

Таблица №2 - Подсчёт сложности реализованного алгоритма

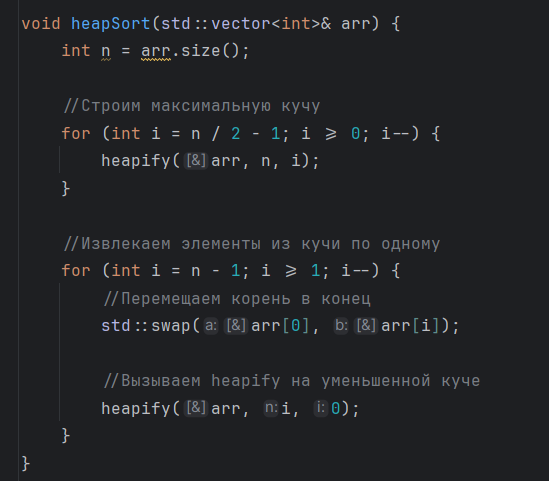
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Размер входного набора | 1000 | 2000 | 4000 | 6000 | 8000 | 10000 |
| Время выполнения программы, с | 0,001 | 0,008 | 0,0013 | 0,02 | 0,023 | 0,05 |

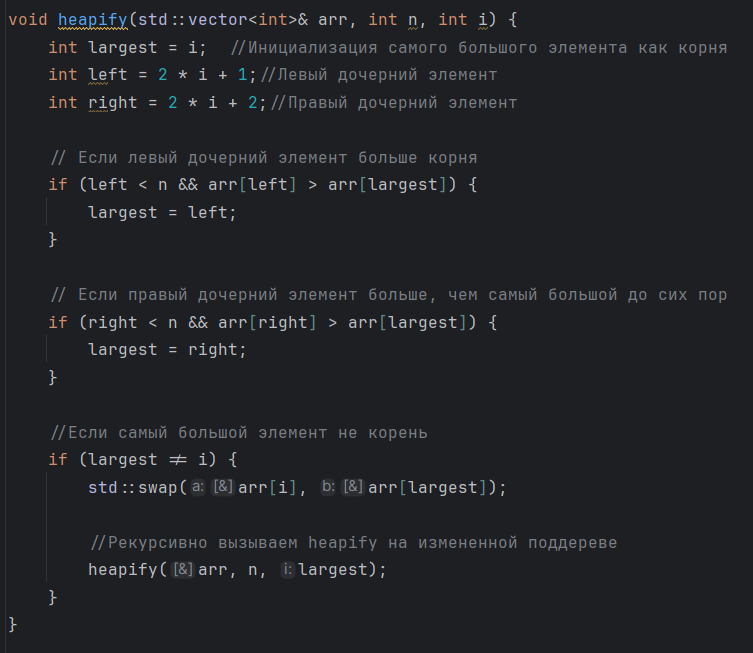
График представляющий визуально удобный формат данных из таблицы №2 представлен на изображении №2.



Изображение №2 - График работы алгоритма

Этот алгоритм гораздо эффективней предыдущего, что заметно на графиках. Также он является отличным выбором сортировки при ограничениях на память.





**Timsort**:

Память: 4n + 2n + log2​(n) \* 20.

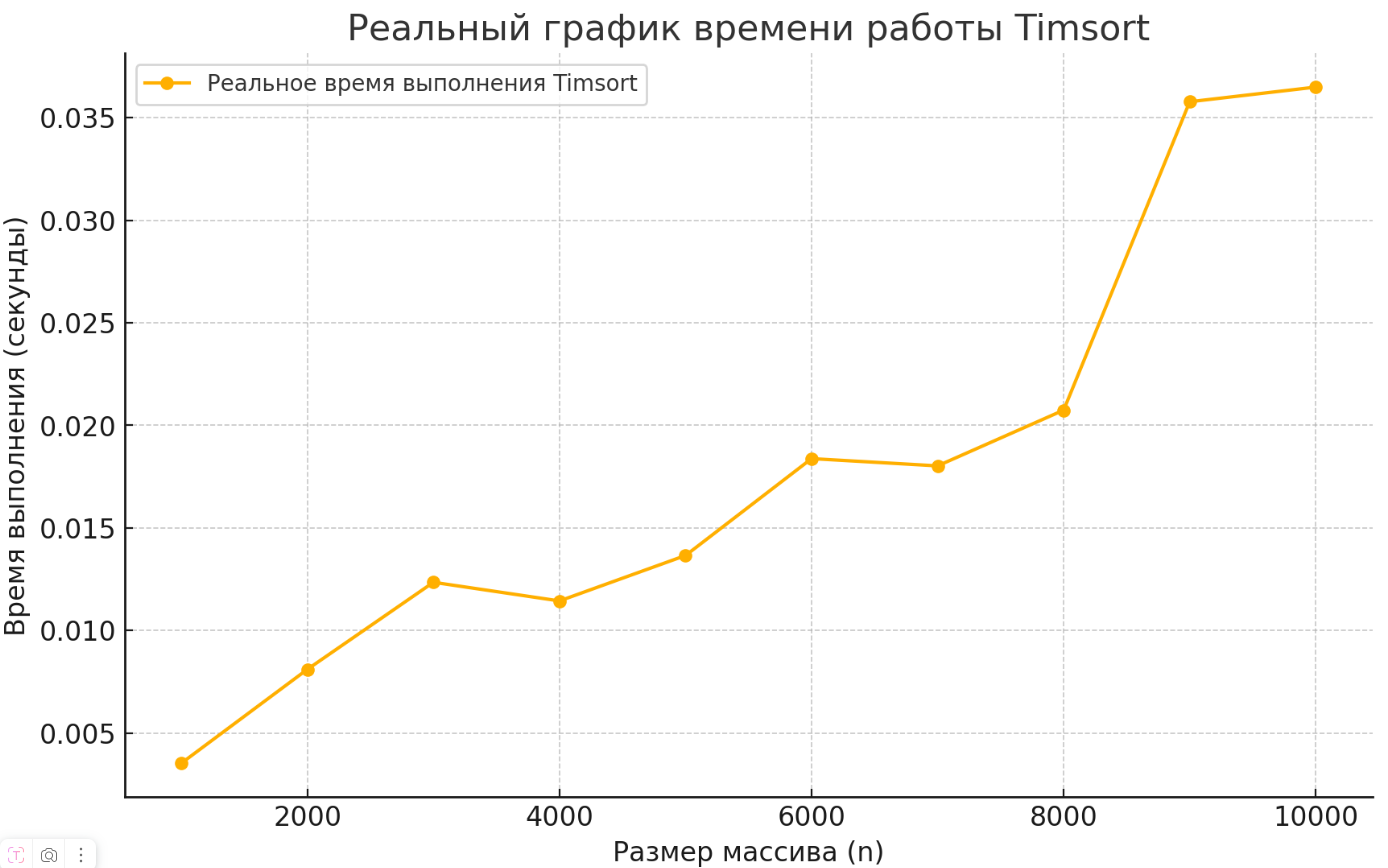
Асимтотика:O(n\*logn)

Согласно требованиям моего варианта, на вход к моему алгоритму подаётся до 1е6 элементов. Теоретически заданная сложность задачи составляет O(n^2). Для тестирования алгоритма была собрана статистика, приведенная в таблице №3.

Таблица №3 - Подсчёт сложности реализованного алгоритма

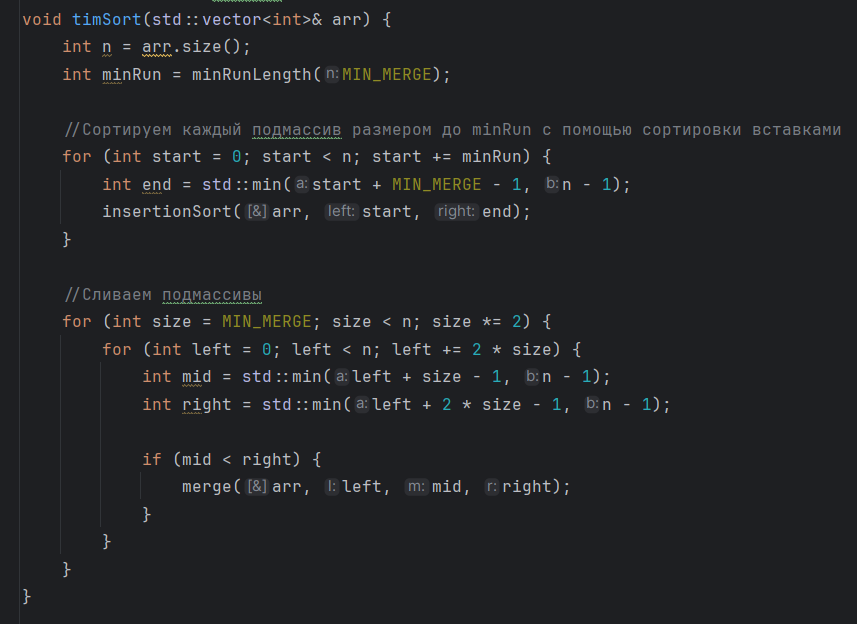
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Размер входного набора | 1000 | 2000 | 4000 | 6000 | 8000 | 10000 |
| Время выполнения, с | 0,0025 | 0,007 | 0,012 | 0,017 | 0,02 | 0,035 |

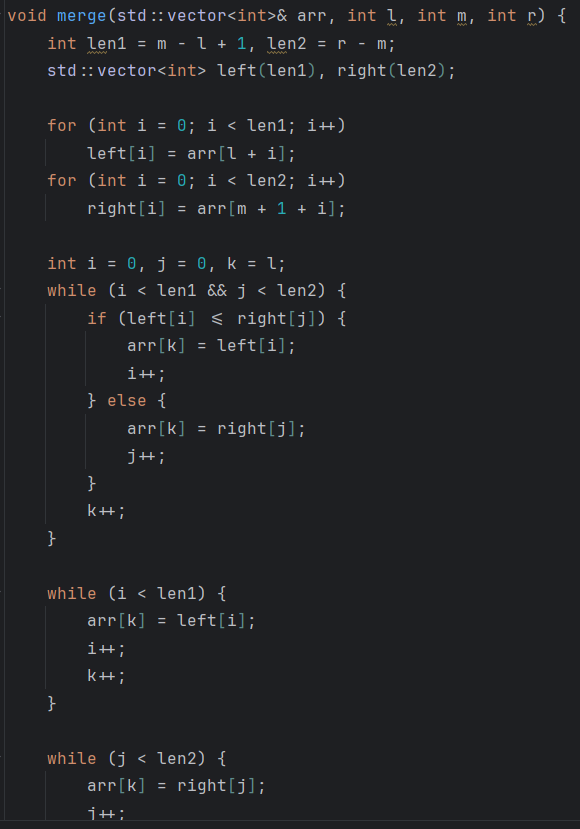
График представляющий визуально удобный формат данных из таблицы №3 представлен на изображении №3.

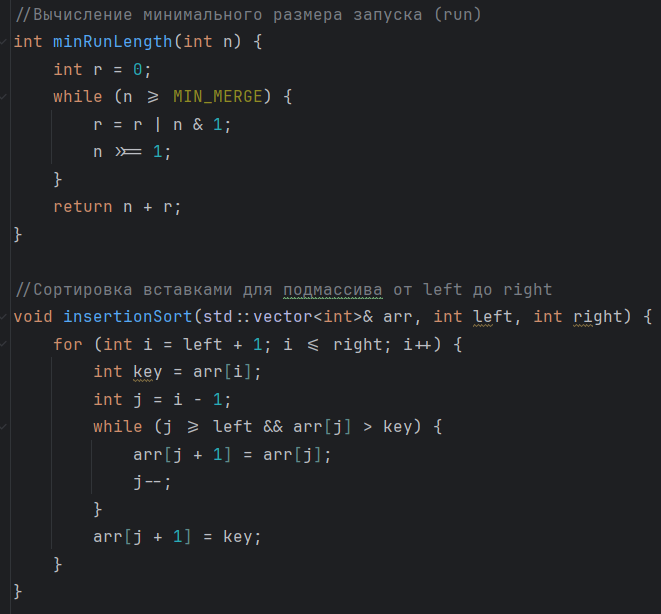


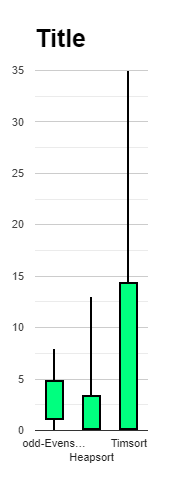
Изображение №3- График работы алгоритма

Этот алгоритм является эффективным засчет частично упорядоченных данных.









Визуализация в Box plot

5.Вывод.

1)Odd-Even Sort O(n^2) непрактичен для больших объемов данных, что подтверждается невозможностью его эффективного выполнения на 100,000 элементов. Его лучше применять только для небольших массивов или в учебных целях для демонстрации алгоритмов сортировки.

2) Heapsort O(n log n) показал схожую асимптотику, но на практике медленнее из-за частых операций перестановки в куче. Он может использоваться в системах с ограниченной памятью, где важно избегать использования дополнительной памяти.

3) Timsort O(n log n) показал на практике лучшие результаты, как и ожидалось, благодаря оптимизациям, встроенным в его реализацию.

Алгоритм подходит для большинства реальных задач благодаря своей эффективности.

ПРИЛОЖЕНИЯ

unittests

