Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский химико-технологический университет имени Д.И. Менделеева»

Кафедра информационных компьютерных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

Выполнил студентка группы КС-36 Битарова Эмма Олеговна

Ссылка на репозиторий: https://github.com/MUCTR-IKT-CPP/

Приняли: Пысин Максим Дмитриевич

Дата сдачи: 27.11.2022

Оглавление

[Описание задачи. 2](#_Toc63548272)

[Описание метода/модели. 2](#_Toc63548273)

[Выполнение задачи. 2](#_Toc63548274)

[Заключение. 2](#_Toc63548275)

# Описание задачи.

Реализовать на выбранном языке метод сортировки слиянием. Продемонстрировать его работоспособность на наборе разработанных самостоятельно тестов. в каждом тесте необходимо по 20 раз генерировать вектор, состоящий из N элементов. При проведении сортировки подсчитать колличество дополнительной потребляемой памяти, под дополнительной понимается память, которая используется на хранение промежуточного результата сортировки.

При работе сортировки подсчитать количество вызовов рекурсивной функции, и высоту рекурсивного стека.

# Описание метода/модели.

# Один из самых быстрых известных универсальных алгоритмов сортировки массивов: в среднем 𝑂(𝑛 log⁡𝑛 ) обменов при упорядочении 𝑛 элементов; из-за наличия ряда недостатков на практике обычно используется с некоторыми доработками. Сортируемый массив разбивается на две части примерно одинакового размера; Каждая из получившихся частей сортируется отдельно, например — тем же самым алгоритмом; Два упорядоченных массива половинного размера соединяются в один.

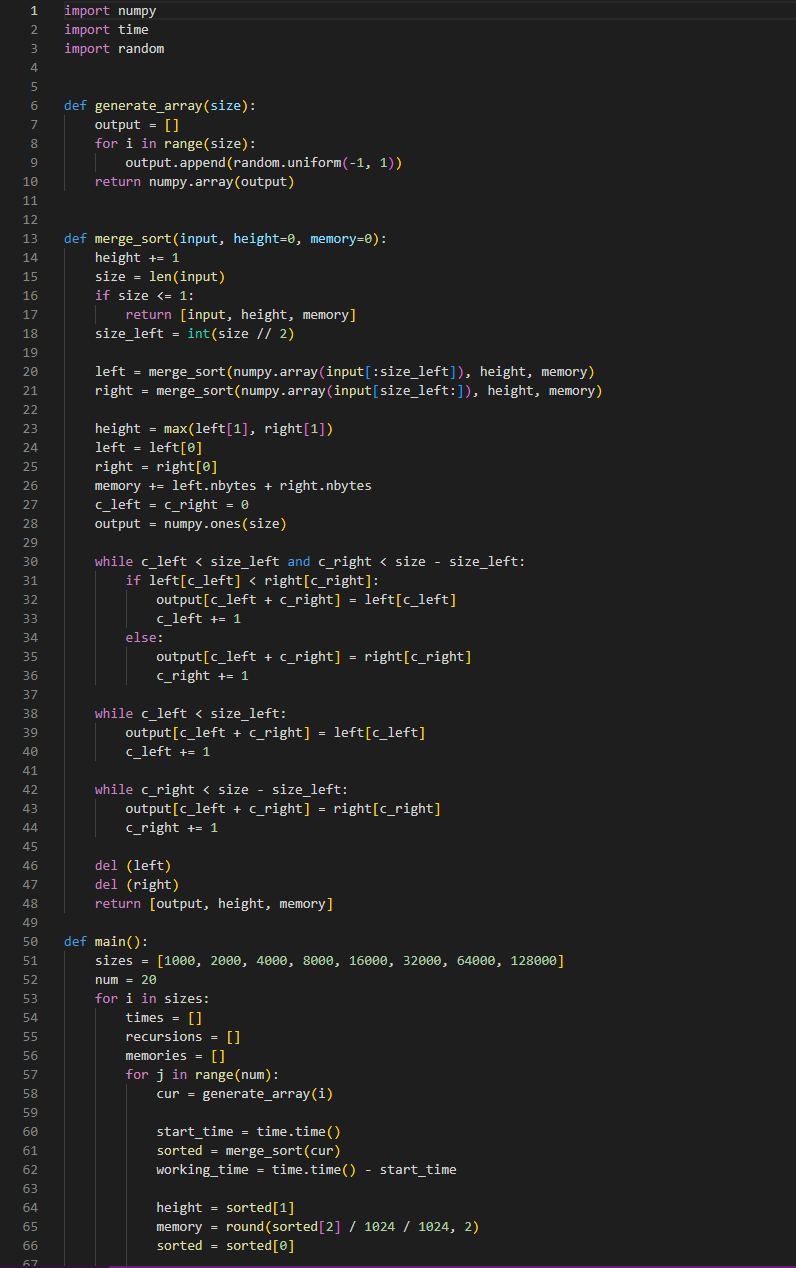
# У нас есть два массива a и b (фактически это будут две части одного массива, но для удобства будем писать, что у нас просто два массива). Нам надо получить массив c размером |a|+|b|. Для этого можно применить процедуру слияния. Эта процедура заключается в том, что мы сравниваем элементы массивов (начиная с начала) и меньший из них записываем в финальный. И затем, в массиве у которого оказался меньший элемент, переходим к следующему элементу и сравниваем теперь его. В конце, если один из массивов закончился, мы просто дописываем в финальный другой массив. После мы наш финальный массив записываем заместо двух исходных и получаем отсортированный участок.

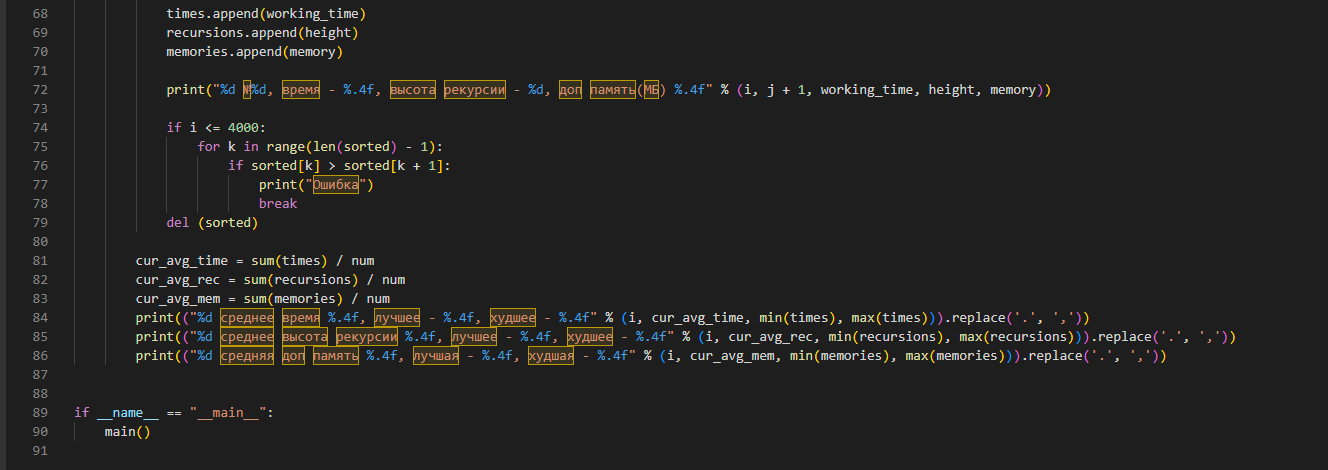
# Выполнение задачи.

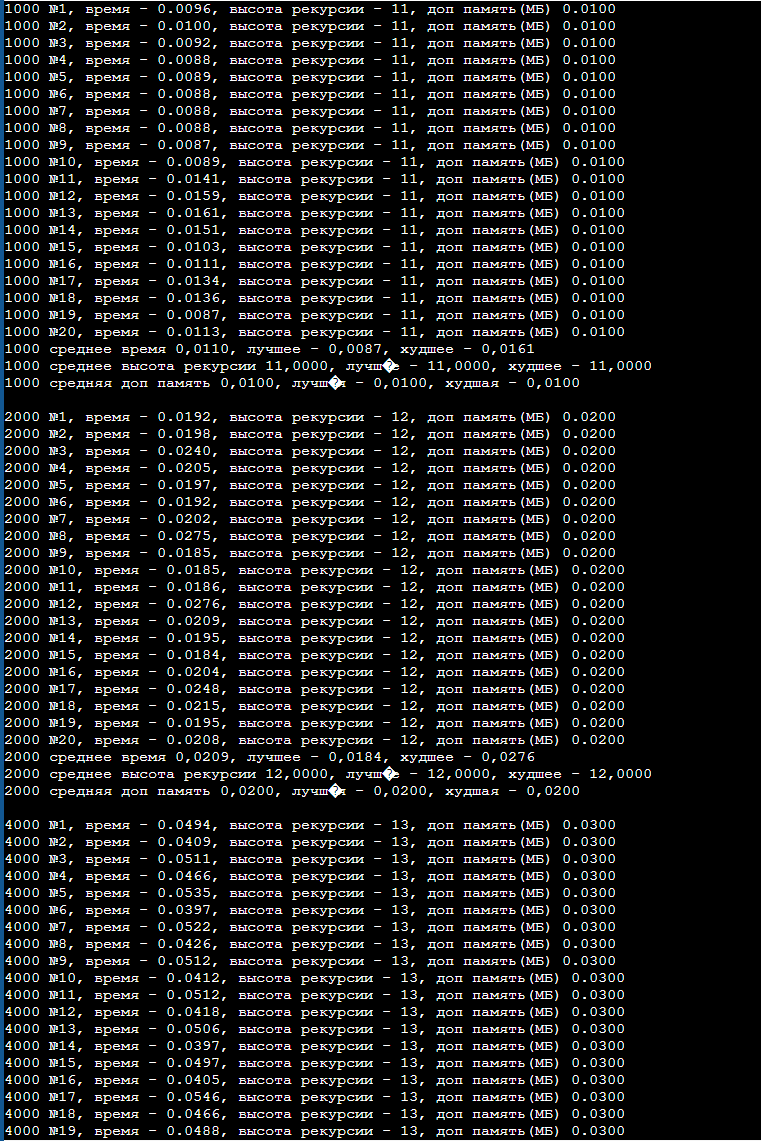
Язык:Python.

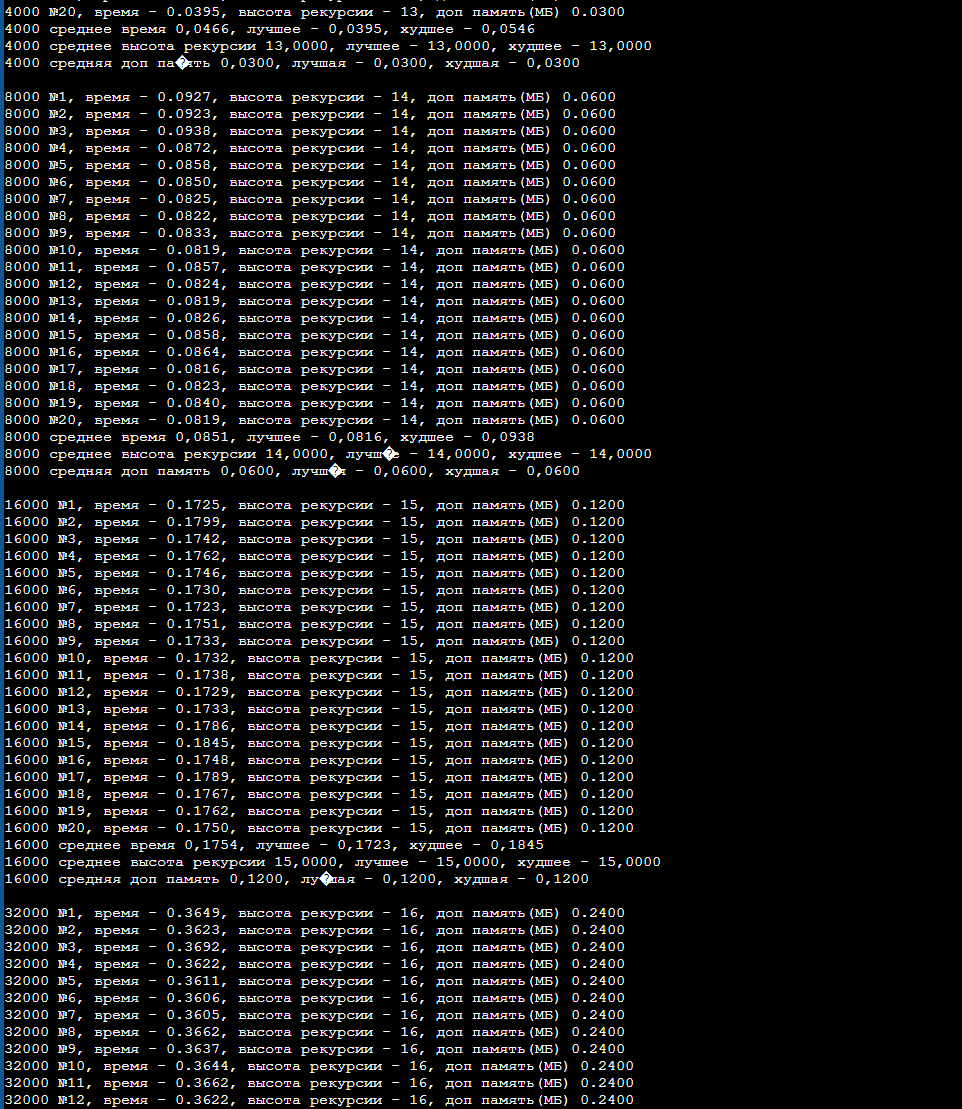
Была написана функция выполнения сортировки слиянием. Массив делился на две части, каждая из частей сортировалась отдельно. В моем случае тем же способом. Сравниваются элементы массива, начиная с начала и меньший записывался в финальный. Затем идет переход к следующему элементу в массиве с меньшим элементом. После его окончания, дописывается другой массив в окончательный массив.

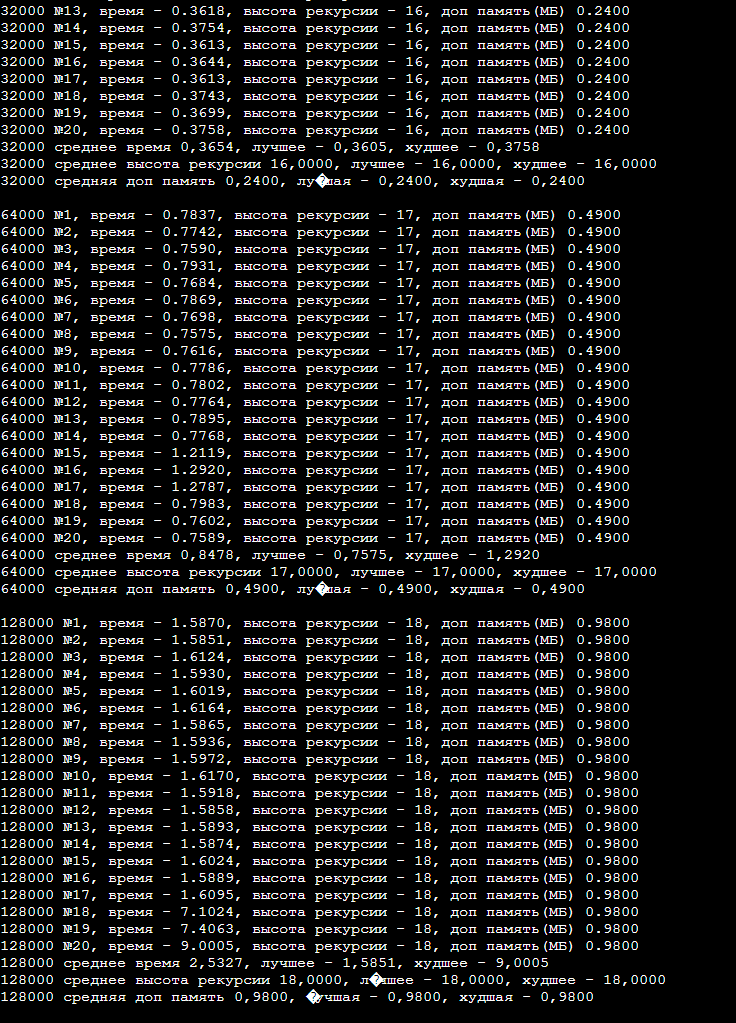
Так же было подсчитано время, которое затрачивалось на сортировку каждого массива, высота рекурсии, среднее время, высота и использованная дополнительная память.











# Заключение.

Данный метод сортировки показывает хорошие результаты на протяжении всей работы, даже при большом размере массива.