Отчет по курсовой работе.
Алгоритм внешней сортировки.

Задача:

Есть некий файл с данными, размер которых превышает объем оперативной памяти. Необходимо отсортировать строки в этом файле.

Алгоритм:

1. Считываем данные с файла в память, пока хватает памяти на хранение и сортировку этих данных.
2. Сортируем часть данных и записываем их в новый файл.
3. Повторяем шаги 1-2 пока не все данные считаны из исходного файла
4. Операция слияния происходит над набором файлов, по количеству не превышающие некоторое число, зависящее от кол-во доступной памяти. Выполняем для набора слияние (пункт 5-7) и записываем в новый файл, пока количество файлов больше одного.
5. Считываем из каждого файла строку, выбираем из всех строк минимальную и записываем в результирующий файл.
6. Для файла, строка которого оказалась наименьшей, считываем следующую, если такая есть.
7. Повторяем шаги 4-5 пока есть строки, не записанные в ответ

Шаг 5 можно ускорить, если для поиска минимума использовать структуру данных - приоритетную очередь.

Шаг 6 также можно ускорить, если для каждого файла хранить небольшой буфер с данными, чтобы не считывать новую строку каждый раз.

Тестирование:

Размер памяти для хранения данных для простоты задавался количеством строк, которое можно одновременно держать в памяти. Результаты приведены в таблице 1

В таблице записано время работы, усредненные за 3 запуска программы. Время измерялось простой разницей времени перед запуском и после, так как приложение работает в одном потоке. Память измерялась с помощью утилит для слежения использования памяти процессами.

Также было проверено время работы для файла размером 25 GB и размером памяти в 100000 строк - 1230 секунд (20 минут 30 секунд)

Размер памяти/размер файла	20 MB	600 MB	1 GB	5 Gb
100	1.3	150	180	984
1000	1.1	81	92	525
50000	1	61	82	337
100000	1	67	73	296

Таблица 1. Зависимость времени(с.) от размера файла и размера буфера(количеством строк)