Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Уфимский государственный нефтяной технический университет»

Факультет автоматизации производственных процессов

Кафедра вычислительной техники и инженерной кибернетики

ОТЧЕТ

по лабораторной работе №2

«Исследование алгоритмов внешних сортировок»

Выполнил ст. гр. БПО-15-02 Фарухьянова Ю.Р.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

Проверил: преподаватель Жолобова Г.Н.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

Целью данной лабораторной работы является: изучение основных методов упорядочивания данных, расположенных во внешней памяти; определение производительности алгоритмов сортировки; применение алгоритмов сортировки последовательностей.

Исследуемые методы внешней сортировки:

1. Простое слияние;
2. Естественное слияние.

Общий алгоритм сортировки слиянием:

Сначала серии распределяются на два или более вспомогательных файлов. Данное распределение идет поочередно: первая серия записывается в первый вспомогательный файл, вторая – во второй и так далее до последнего вспомогательного файла. Затем опять запись серии начинается в первый вспомогательный файл. После распределения всех серий, они объединяются в более длинные упорядоченные отрезки, то есть из каждого вспомогательного файла берется по одной серии, которые сливаются. Если в каком-то файле серия заканчивается, то переход к следующей серии не осуществляется. После того как все серии из всех вспомогательных файлов объединены в новые серии, потом опять начинается их распределение. И так до тех пор, пока все данные не будут отсортированы.

Алгоритм сортировки простым слиянием:

В данном алгоритме длина серий фиксируется на каждом шаге. В исходном файле все серии имеют длину 1, после первого шага она равна 2, после второго – 4, после третьего – 8, после k-го шага – 2k.

1. Шаг 1. Исходный файл f разбивается на два вспомогательных файла f1 и f2;
2. Вспомогательные файлы f1 и f2 сливаются в файл f, при этом одиночные элементы образуют упорядоченные пары;
3. Полученный файл f вновь обрабатывается, как указано в шагах 1 и 2. При этом упорядоченные пары переходят в упорядоченные четверки;
4. Повторяя шаги, сливаем четверки в восьмерки и т.д., каждый раз удваивая длину слитых последовательностей до тех пор, пока не будет упорядочен целиком весь файл.

Алгоритм сортировки естественным слиянием:

В случае простого слияния частичная упорядоченность сортируемых данных не дает никакого преимущества. Это объясняется тем, что на каждом проходе сливаются серии фиксированной длины. При естественном слиянии длина серий не ограничивается, а определяется количеством элементов в уже упорядоченных подпоследовательностях, выделяемых на каждом проходе.

1. Исходный файл f разбивается на два вспомогательных файла f1 и f2. Распределение происходит следующим образом: поочередно считываются записи ai исходной последовательности (неупорядоченной) таким образом, что если значения ключей соседних записей удовлетворяют условию f(ai)<=f(ai+1), то они записываются в первый вспомогательный файл f1. Как только встречаются f(ai)>f(ai+1), то записи ai+1копируются во второй вспомогательный файл f2. Процедура повторяется до тех пор, пока все записи исходной последовательности не будут распределены по файлам;
2. Вспомогательные файлы f1 и f2 сливаются в файл f, при этом серии образуют упорядоченные последовательности;
3. Полученный файл f вновь обрабатывается, как указано в шагах 1 и 2;
4. Повторяя шаги, сливаем упорядоченные серии до тех пор, пока не будет упорядочен целиком весь файл.

Для реализации сравнения эффективности указанных сортировок и наглядности полученных результатов реализуем приложение с графическим интерфейсом в среде программирования QtCreator.

Определимся с графическим интерфейсом пользователя. В окне приложения должна быть реализована возможность выбора метода внешней сортировки. Также должны присутствовать 2 окна для вывода текстовой информации: содержимое файла до сортировки и содержимое файла после сортировки. Добавим 2 кнопки: Загрузить, используемую для отображения данных из входного файла, и Сортировать, используемую для выполнения сортировки данных и отображения результатов в окне приложения. Для сравнения эффективности методов сортировки добавим небольшую панель для отображения результатов, куда будут выводиться сведения о количестве отсортированных элементов и времени сортировки.

Ниже представлена реализация созданного приложения при n = 39999.

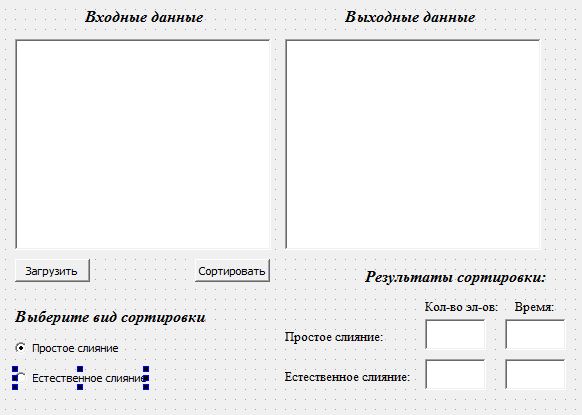


Рисунок 1 – Окно графического интерфейса пользователя

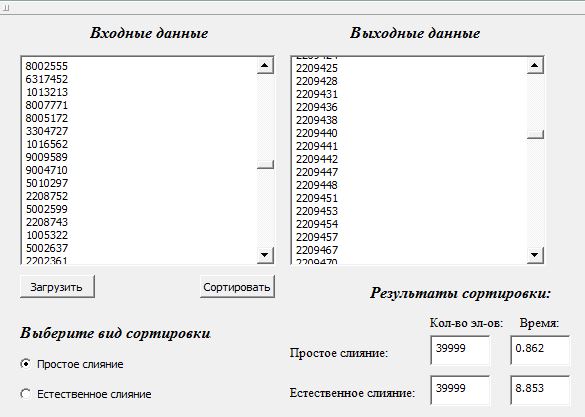


Рисунок 2 – Реализация сортировок

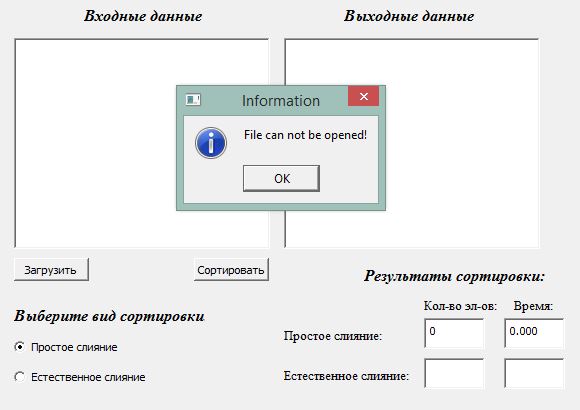


Рисунок 3 – Окно приложения при отсутствующем входном файле

Был проведен анализ времени выполнения сортировки при различных n и построены графики зависимости t,ms от N (O-notation). Результаты можно увидеть на рисунках 6, 7.

|  |  |  |
| --- | --- | --- |
|  | Простое | Естественное |
| n | t1 | t2 |
| 39999 | 0,862 | 8,853 |
| 35000 | 0,718 | 7,662 |
| 32679 | 0,608 | 6,73 |
| 29047 | 0,567 | 5,987 |
| 26631 | 0,512 | 5,459 |
| 21649 | 0,458 | 4,451 |
| 19194 | 0,39 | 3,968 |
| 17390 | 0,376 | 3,567 |

Рисунок 6 – О-нотация сортировки простым слиянием

Рисунок 7 – О-нотация сортировки естественным слиянием

Вывод:

В ходе выполнения данной лабораторной работы были изучены методы внешних сортировок. Сравнивая полученные результаты, можно увидеть, насколько различается время выполнения сортировок простым и естественным слиянием, но при этом зависимости времени выполнения сортировки от количества элементов во входном файле не сильно отличаются. Время выполнения сортировки зависит от нескольких факторов: во-первых, как упорядочены элементы во входном файле (лучший/худший случаи), на сколько файлов разбивается исходная последовательность, фиксирована/не фиксирована длина серии и сколько выполняется сравнений. Число чтений или перезаписей файлов при использовании метода естественного слияния будет не хуже, чем при применении метода простого слияния, а в среднем – даже лучше. Но в этом методе увеличивается число сравнений за счет тех, которые требуются для распознавания концов серий, именно поэтому времени затрачивается много больше.