

Лабораторная работа №5. «Сортировка слиянием»

1.1. Цель работы

Ознакомиться и реализовать алгоритм сортировки слиянием.

1.2. Сортировка слиянием

Слияние – объединение двух отсортированных подмассивов в один. Таким образом, слияние сортирует два подмассива и сливает их для получения отсортированного массива.

Слияние не зависит от характера входных данных. Основной недостаток метода – необходимость дополнительного объема памяти.

Сортировка слиянием используется в следующих случаях:

- Быстродействие выходит на первое место
- Легко применяется для структур с последовательным доступом к данным
- При решении задач следующего типа – основной набор данных уже отсортирован, в систему постоянно поступают новые данные, которые необходимо разместить, не теряя упорядоченности всего набора данных.

1.3. Способы программирования сортировки

1.3.1. Нисходящая сортировка слиянием

Первый способ программирования сортировки очень напоминает способ программирования сортировки Хоара. Функция сортировки рекурсивная, зависит от сортируемого массива и границ сортируемого массива. Запишем алгоритм сортировки:

Используемые обозначения:

X - сортируемый массив

F - индекс начала массива

L - индекс конца массива

Сортировка (X, F, L)

1. ЕСЛИ $L \leq F$ ТО закончить работу
2. $M = (L+F)/2$ // найти середину
3. *Сортировка*(X, F, M) // вызвать сортировку для левого
// подмассива
4. *Сортировка*($X, M+1, L$) // вызвать сортировку для правого
// подмассива
5. *Слияние* ($X, F, M+1, L$)

На рис. 1.1 изображено дерево разбиений, которое строит вышеописанный алгоритм на массиве из 25 элементов (элементы дерева – количество подмассивов).

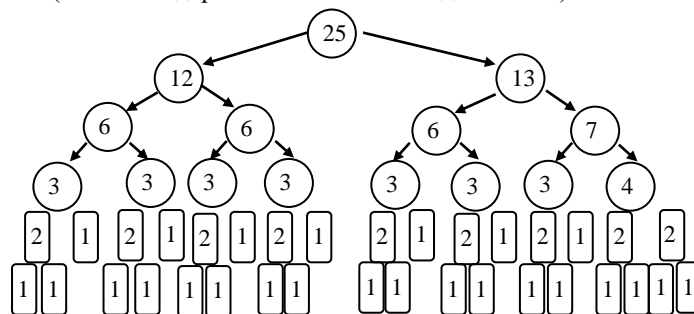


Рис. 1.1. Дерево разбиений, построенное нисходящим слиянием.

Как только алгоритм выполнил разбиение на подмассивы по одному элементу (нижний уровень дерева) начинает свою работу алгоритм слияния.

1.3.2. Восходящая сортировка слиянием

Дерево разбиений, построенное алгоритмом восходящей сортировки, представлено на рис. 1.2. Восходящая сортировка слиянием просматривает массив слева направо, и на первом шаге сливает рядом стоящие одиночные элементы в упорядоченные пары элементов. На втором шаге – рядом стоящие упорядоченные пары в упорядоченные четверки элементов. И так далее. Алгоритм восходящей сортировки

не рекурсивен. Функция, выполняющая слияние вызывается сразу же после выделения границ объединяемых подмассивов.

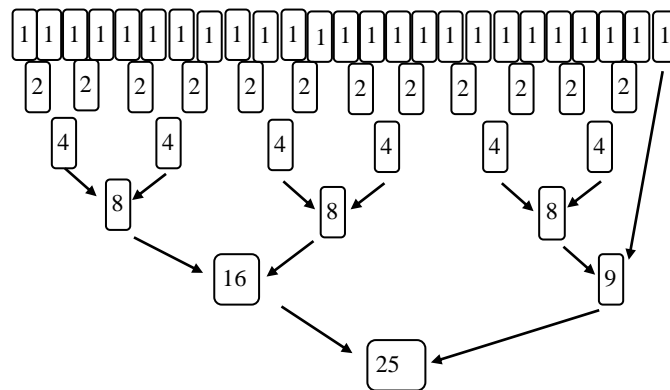


Рис. 1.2. Дерево восходящей сортировки слиянием.

```

Для  $h$  от 1 до  $n$  шаг
  Для  $i$  от 0 до  $n - h$  шаг
    Если  $(i + 2 * h - 1) < n$  то      merge( $X, i, i + h, i + h * 2 - 1$ )
    иначе merge( $X, i, i + h, n - 1$ )
  шаг
h =  $h * 2$ ;
конеч

```

1.4. Слияние подмассивов

1.4.1. Прямое (двухпутевое) слияние.

Алгоритм прямого слияния можно описать следующим образом:

пусть F – начало первого подмассива, L – конец второго подмассива, M – начало второго подмассива, X – массив.

```

Merge( $X, F, M, L$ )
  начало
  Определить массив  $X1$ 
   $i = F, j = M, k = 0$ 
  Пока  $(i < M$  и  $j <= L)$ 
    Если  $X[i] > X[j]$  то       $X1[k] = X[j]$ 
                            $j = j + 1$ 
                            $k = k + 1$ 
    Иначе Если  $(X[j] > X[i])$  То  $X1[k] = X[i]$ 
                            $i = i + 1$ 
                            $k = k + 1$ 
    Иначе  $X1[k] = X[i]$ 
         $X1[k + 1] = X[j]$ 
         $k = k + 2$ 
         $i = i + 1, j = j + 1$ 
  шаг
  Если  $i < M$  то дописать элементы
   $X[i], \dots, X[M - 1]$  в  $X1$ 
  Если  $j <= L$  то дописать элементы
   $X[j], \dots, X[L]$  в  $X1$ 
  Переписать элементы из  $X1$  в  $X$ 
  Конеч

```

1.4.2. Абстрактное обменное слияние

При программировании прямого слияния необходимо постоянно отслеживать концы сливаемых подмассивов и случай равенства просматриваемых элементов.

Можно избавиться от этих недостатков, применяя следующую схему слияния:

1. Объединить первый и второй подмассивы во вспомогательный по следующему правилу:
 - элементы первого подмассива переписываем в прямом порядке;
 - элементы второго подмассива переписываем в обратном порядке;
2. Используем индекс i - для первого подмассива (изменяем его слева направо во вспомогательном массиве) и индекс j - для просмотра второго подмассива, изменяем его справа налево во вспомогательном массиве.
3. Сравним элементы, находящиеся на позициях i, j и переносим в результирующий массив меньший из элементов.

Merge2(X,F,M,L)

начало

Определить массив X1

Переписать элементы из массива X в массив X1 по правилу битонной последовательности

$$i=0, j = L-F, k=F$$

Пока ($i < j$)

```

Если X1[i]>X1[j] то      X[k]=X1[j]
                          j=j-1

```

Иначе $X[k]=X1[i]$
 $i=i+1$

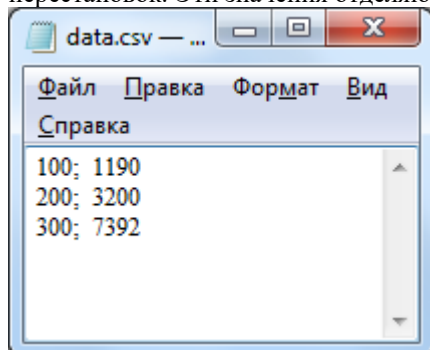
$$k=k+1$$

КЦ

Конец

1.5. Порядок выполнения

- Получите вариант индивидуального задания у преподавателя.
 - Составьте алгоритм сортировки.
 - Реализуйте алгоритм на языке Си, добавив в программу подсчет количества сравнений и перестановок, проведенных алгоритмом.
 - **Обратите внимание!!! Так же, как и в двух предыдущих работах** - При сдаче лабораторной работы должно быть написано **две** программы:
1. Программа, реализующая сортировку по индивидуальному заданию для массива произвольной размерности.
 2. Программа, которая последовательно запускается для массива из 100 элементов **один** раз, и подсчитывает количество сравнений и перестановок на массиве из 100 элементов, для массива из 200 элементов **один** раз и подсчитывает количество сравнений и перестановок на массиве из 200 элементов, и так далее, увеличивая размерность массива до 10000 элементов. **Подумайте, почему** при подсчете сравнений и перестановок в сортировке слиянием не нужно вычислять среднее значение сравнений и перестановок?
 3. Для удобства конвертации полученного текстового файла в Excel или Libre Calc вторая программа должна создавать текстовый файл с расширением *.csv, на каждой строке которого пишется размерность обрабатываемого массива и полученное среднее значение количества сравнений и перестановок. Эти значения отделяются друг от друга точкой с запятой.



- Добавьте полученные данные в файл с данными трех предыдущих лабораторных работ.
- Постройте графики по данным четырех занятий на одном поле.

Сделайте выводы по работе.