Лекция 11

Сортировка массивов

Основные методы сортировки

Сортировка – процесс перегруппировки заданного множества объектов в соответствии определенным критерием (например, по возрастанию).

Цель сортировки – облегчить последующий поиск элементов в отсортированном множестве.

Методы сортировки:

- ***внутренней сортировки*** (не предусматривают использование дополнительных массивов, массивы полностью располагаются в ОП);
- ***внешней сортировки*** (применяются к большим массивам, расположенным на внешних носителях).

Элементарные методы внутренней сортировки :

- сортировка вставкой (включением);
- сортировка выбором;
- сортировка обмен (пузырьковая сортировка).

Усовершенствованные методы внутренней сортировки:

- быстрая сортировка (метод Хоара);
- сортировка методом Шелла;
- сортировка с помощью дерева или пирамидальная сортировка;
- сортировка методом слияния.

Требования к алгоритму сортировки массивов

Требования к алгоритму сортировки массивов:

- **устойчивость** (относительное расположение элементов с равными значениями не меняется);
- **экономное использование памяти** (сортировка должна выполняться на том же месте);
- **эффективность** (время работы).

Мерой эффективности является:

C – число операций сравнения;

M – число операций обмена (пересылок данных).

Эти числа являются функцией от числа элементов массива **N.**

Хорошие алгоритмы сортировки требуют порядка $\mathbf{N}*\log_2\mathbf{N}$ операций сравнения.

Сортировка методов вставки (включения)

Элементы массива мысленно делятся на две группы: ***уже отсортированную*** последовательность **a[0], …, a[i-1]** и ***неотсортированную*** последовательность. На каждом шаге, начиная с **i=1**, из ***неотсортированной*** последовательности выбирается **i-ый** элемент и вставляется в ***уже отсортированную*** последовательность на нужное место.

1-й этап	38 |	(12)	80	15	36	23	74	62
2-й этап	12	38 |	(80)	15	36	23	74	62
3-й этап	12	38	80 |	(15)	36	23	74	62
4-й этап	12	15	38	80 |	(36)	23	74	62
5-й этап	12	15	36	38	80 |	(23)	74	62
6-й этап	12	15	23	36	38	80 |	(74)	62
7-й этап	12	15	23	36	38	74	80 |	(62)
	12	15	23	36	38	62	74	80

Число операций сравнения $C_{\max} = (n^2 + n - 4)/4$

Число операций обмена $M_{\max} = (n^2 + 3n - 4)/2$

Сортировка методов вставки (включения)

1-й этап	38 |	(12)	80	15	36	23	74	62
2-й этап	12	38 |	(80)	15	36	23	74	62
3-й этап	12	38	80 |	(15)	36	23	74	62
4-й этап	12	15	38	80 |	(36)	23	74	62
5-й этап	12	15	36	38	80 |	(23)	74	62
6-й этап	12	15	23	36	38	80 |	(74)	62
7-й этап	12	15	23	36	38	74	80 |	(62)
	12	15	23	36	38	62	74	80

Фрагмент программы:

```
const int N=8; // размер массива
 int i,j,      // индексы
     temp,     // временная переменная
 a[N]={38, 12, 80, 15, 36, 23, 74, 62}; // массив
 // сортировка
  for(i=1;i<N;i++){
    temp=a[i];                              // запоминание
    for(j=i-1;(j>=0)&&(a[j]>temp);j--)      // поиск места вставки
        a[j+1]=a[j];                        // сдвиг эл. массива
    a[j+1]=temp;                            // вставка
   /* cout<<endl; //тестовая печать
    for(k=0;k<N;k++) cout<<" "<<a[k];*/
  }
```

```
12 38 80 15 36 23 74 62
12 38 80 15 36 23 74 62
12 15 38 80 36 23 74 62
12 15 36 38 80 23 74 62
12 15 23 36 38 80 74 62
12 15 23 36 38 74 80 62
12 15 23 36 38 62 74 80
```

Сортировка методом прямого выбора

1. Выбирается наименьший элемент в неотсортированной части.
2. Он меняется местами с последним элементом в уже отсортированной части

Начальное состояние массива	8	23	5	65	44	33	1	6
Шаг 1	1	23	5	65	44	33	8	6
Шаг 2	1	5	23	65	44	33	8	6
Шаг 3	1	5	6	65	44	33	8	23
Шаг 4	1	5	6	8	44	33	65	23
Шаг 5	1	5	6	8	23	44	65	33
Шаг 6	1	5	6	8	23	33	65	44
Шаг 7	1	5	6	8	23	33	44	65
	1	5	6	8	23	33	44	65

Алгоритм:

```
for (i=0;i<N;i++){
    присвоить imin индекс наименьшего из a[i]… a[n]
              temp наименьшее значение из a[i]… a[n];
    поменять местами a[imin] и a[i];
}
```

Число операций сравнения $C_{\max} = (n^2 - n)/2$

Число операций обмена $M_{\max} = n^2/4 + 3(n-1)$

Сортировка методом прямого выбора

```
const int N=8;// размер
int i, j,     // индексы
      a[N],   // массив
      temp,   // минимум
      imin;   // индекс минимума
// сортировка
  for (i=0;i<N;i++){
      imin=i; temp=a[i];
      for(j=i+1;j<N;j++)   // поиск мин. эл-та
          if(a[j]<temp){
            imin=j;
            temp=a[imin];
            }
      a[imin]=a[i];   //перестановка элементов
      a[i]=temp;
      /* cout<<endl; //тестовая печать
      for(k=0;k<N;k++) cout<<" "<<a[k];*/
   }
```

8	23	5	65	44	33	1	6
1 |	23	5	65	44	33	8	6
1	5|	23	65	44	33	8	6
1	5	6 |	65	44	33	8	23
1	5	6	8 |	44	33	65	23
1	5	6	8	23 |	44	65	33
1	5	6	8	23	33 |	65	44
1	5	6	8	23	33	44 |	65
1	5	6	8	23	33	44	65 |

```
1 23 5 65 44 33 8 6
1 5 23 65 44 33 8 6
1 5 6 65 44 33 8 23
1 5 6 8 44 33 65 23
1 5 6 8 23 33 65 44
1 5 6 8 23 33 65 44
1 5 6 8 23 33 44 65
1 5 6 8 23 33 44 65
```

Сортировка прямым обменом

Алгоритм основан на сравнении и смене мест двух соседних элементов и продолжении этого процесса до тех пор, пока не будут упорядочены все элементы.

При каждом проходе по массиву наименьший элемент сдвигается к левому концу массива. Наименьший элемент можно интерпретировать как «пузырек», который поднимается до уровня, соответствующего его весу.

17	**42**	**66**	**68**	**32**	**13**	**55**	**11**
17	42	66	32	13	55	11	68
17	42	32	13	55	11	66	68
17	32	13	42	11	55	66	68
17	13	32	11	42	55	66	68
13	17	11	32	42	55	66	68
13	11	17	32	42	55	66	68
11	13	17	32	42	55	66	68

Число операций сравнения $C_{\max} = (n^2 - n)/2$

Число операций обмена $M_{\max} = 3(n^2 - n)/4$

Сортировка прямым обменом

```
const int N=8;
int i,j,k,    // индексы
    a[N],     // массив из целых чисел
    temp;     // временная переменная
// сортировка
for (j=N; j>1; j--) {
  for(i=0; i<j-1; i++) // сравнение двух соседних элементов
      if(a[i]>a[i+1]) {
            temp=a[i];  // перестановка элементов
            a[i]=a[i+1];
            a[i+1]=temp;
      }
//тестовая печать
for(k=0; k<N; k++) cout<<" "<<a[k];
cout<<endl;
}
```

```
17 42 66 32 13 55 11 68
17 42 32 13 55 11 66 68
17 32 13 42 11 55 66 68
17 13 32 11 42 55 66 68
13 17 11 32 42 55 66 68
13 11 17 32 42 55 66 68
11 13 17 32 42 55 66 68
```