

Лабораторная работа №4.6.9
по дисциплине «Информатика и программирование»
Сортировка и поиск

Группа: **АВТ-010**

Студенты: **Антонов Н., Довиденко Р..**

Преподаватель: **Романов Е.Л.**

НОВОСИБИРСК 2010

Задание

Сортировка «хитрая». Из массива путем однократного просмотра выбирается последовательность элементов, находящихся в порядке возрастания, переносятся в выходной массив и заменяются во входном на **-1**. Затем оставшиеся элементы включаются в полученную упорядоченную последовательность методом погружения.

Теоретические сведения

Тривиальные сортировки (в т.ч. погружение) имеют трудоемкость вида N^2 . Требуется подсчитать количество погружений и оценить трудоемкость для новой сортировки по этому параметру.

Проектирование программы

Обсуждение основных идей алгоритма

Идея: Из последовательности выделяется уже упорядоченная и переносится в выходной массив, заменяясь во входном на -1. Затем элементы из входного переносятся по одному в выходной, погружением входя на свои места по возрастанию.

«Составные части» программы

Цикл поиска возрастающей последовательности

```
for (i=0; i<n; i++, u++)
{
    if (c<in[i]) c=in[i], out[j++]=c, in[i]=-1, m++;
}
```

Цикл погружения оставшихся элементов

```
for (i=0; i<n; i++)
{if(in[i]!=-1) {
    out[t]=in[i];
    for ( int k=t; k !=0 && out[k] < out[k-1]; k--, u++){
        c=out[k]; out[k]=out[k-1]; out[k-1]=c; b++;}
    t++;
}
}
```

Переменные:

i, j, k – счетчики.

a, in – входной массив.

b, out – выходной массив.

c – временная переменная (для «трех стаканов»).

u – количество шагов цикла.

m – количество перемещенных элементов

(из возрастающей последовательности).

s – количество погружений.

Текст программы с комментариями

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

void sort_main(int in[],int n, int out[])
{
    int j=1,c,i,t,s,m,u;
```

```

c=out[0]=in[0];
in[0]=-1;
u=0;
m=0;
s=0;
for (i=0; i<n; i++, u++)
{
    if (c<in[i]) c=in[i], out[j++]=c, in[i]=-1, m++;
    //находим возрастающую последовательность и переносим ее
    .....
printf("%d ", s); //количество погружений
printf("%d ", m); //количество перемещенных
printf("%d \n", u); //количество шагов цикла
}

void main()
{
    const int n=1000;
    int i, a[n], b[n];
    srand(time(NULL));
    for (i=0; i<n; i++) a[i]=rand()%1000;
    sort_main(a,n,b);
    for (i=0; i<n; i++) printf("%d ",b[i]);
    puts("");
}

```

Пример работы программы

```

260613 4 261613
251565 4 252565
250379 11 251379
246941 8 247941
257314 5 258314
248288 3 249288
254097 11 255097
254821 7 255821
243982 7 244982
256235 7 257235

```

- Число погружений
- Длина изначальной возрастающей последовательности
- Количество шагов цикла

Программа проделывает сортировку 10 раз на различных случайных данных и выводит данные о числе погружений, перемещенной возрастающей последовательности и количестве шагов цикла. Из чего наблюдаем, что на 1000 случайных значений только от 3 до 11 из них есть возрастающая последовательность.

Выводы:

Данная сортировка не дает в существенного увеличения производительности. Из всех элементов только около 1% (от 4 до 11) из 1000 являются возрастающей последовательностью и переносятся сразу же. При значении $N=1000$ получаем $N^2/4=250000$, что практически совпадает с подсчитанной трудоемкостью. «Хитрость» не удалась.