

数据结构第 六 次上机作业

班 级 信嵌（1）班

姓 名

学 号

实验六 排序

1. 实验目的
2. 学习并了解各种排序方法；
3. 比较各种排序算法的优劣；
4. 实验内容
5. 从键盘上输入一组数字，对这组数字用希尔排序，使得这组数字升序排列，将过程和结果显示在屏幕上。
6. 从键盘上输入一组数字，对这组数字用冒泡排序，使得这组数字升序排列，将过程和结果显示在屏幕上。
7. 从键盘上输入一组数字，对这组数字用简单选择排序，使得这组数字升序排列，将过程和结果显示在屏幕上。
8. 实验程序

1.

#include <stdio.h>

void shellsort(int arr[],int length);

int main(int argc, char \*argv[])

{

int arr[]={

13,14,94,33,82,25,59,94,65,23,45,27,73,25,39

};

shellsort(arr,15);

int i=0;

for(;i<15;i++){

printf("%d ",arr[i]);

}

return 0;

}

void shellsort(int arr[],int length){

int gap=length/2;

for(;gap>=1;gap=gap/2){

int m;

for(m=0;m<gap;m++){

int i;

int temp;

for(i=m+gap;i<length;i=i+gap){

int j=i;

while(j!=m){

if(arr[j]<=arr[j-gap]){

temp=arr[j-gap];

arr[j-gap]=arr[j];

arr[j]=temp;

}

j=j-gap;

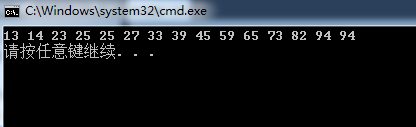
}

}

}

}

}



2.

#include<stdio.h>

#include<stdlib.h>

void BubbleSort(int arr[],int size){

int i,j,k,t;

for(i = 0; i < size-1; i++){

for(j = 0; j < size-1;j++){

if(arr[j] > arr[j +1]){

t = arr[j];

arr[j] = arr[j+1];

arr[j+1] = t;

}

}

}

}

int main(){

int arr[] = {6,4,8,1,2,3,9};

int length;

int i, j;

length = sizeof(arr)/sizeof(arr[0]);

printf("排序前：\n");

for(i = 0; i<length;i++){

printf("%d ",arr[i]);

}

BubbleSort(arr,length);

printf("\n排序后：\n");

for(j = 0; j < length; j++){

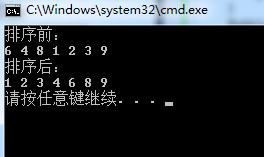
printf("%d ",arr[j]);

}

printf("\n");

return 0;

}



3.

#include <stdio.h>

#include <stdlib.h>

#define MAX 9

typedef struct

{

int key;

}SqNote;

typedef struct

{

SqNote r[MAX];

int length;

}SqList;

void swap(SqNote \*a, SqNote \*b)

{

int key = a->key;

a->key = b->key;

b->key = key;

}

int SelectMinKey(SqList \*L, int i)

{

int min = i;

while (i+1<L->length)

{

if (L->r[min].key>L->r[i+1].key)

{

min=i+1;

}

i++;

}

return min;

}

void SelectSort(SqList \* L)

{

for (int i=0; i<L->length; i++)

{

int j = SelectMinKey(L, i);

if (i != j)

{

swap(&(L->r[i]), &(L->r[j]));

}

}

}

int main()

{

SqList \*L = (SqList\*)malloc(sizeof(SqList));

L->length = 8;

L->r[0].key = 49;

L->r[1].key = 38;

L->r[2].key = 65;

L->r[3].key = 97;

L->r[4].key = 76;

L->r[5].key = 13;

L->r[6].key = 27;

L->r[7].key = 49;

SelectSort(L);

for (int i=0; i<L->length; i++)

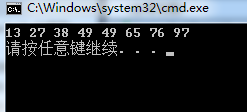
{

printf("%d ",L->r[i].key);

}

return 0;

}



1. 实验反思

①希尔排序算法是先将整个待排记录序列分割成为若干子序列分别进行直接插入排序，待整个序列中的记录“基本有序”时，再对全体记录进行一次直接插入排序。

②冒泡排序算法是每趟排序过程中通过两两比较相邻元素，将小的数字放到前面，大的数字放到后面（升序）。

③简单排序算法是对于具有 n 个记录的无序表遍历 n-1 次，第 i 次从无序表中第 i 个记录开始，找出后序关键字中最小的记录，然后放置在第 i 的位置上。