习题6 排序

6-1 D

6-2 C

6-3 B

6-4 D

6-5 C

6-6 D

6-7

```
//采用计数排序来避免比较和移动
int a[10001],j=0;//j指向数组头
int r[1001];//r中存放1000个数据
for(int i=0;i<1000;++i)
    a[r[i]-1]=r[i];//互不相等可以直接存放
for(int i=0;i<10000;++i)
    if(a[i]>0)
        r[j++]=a[i];//此时存入的数据便由小到大排好
```

6-8

```
struct Node
{
    Type Key;
    Node *next;
};
void ListSort(Node *head,int n)//插入排序
{
    Node *sorted_list = NULL, *p = sorted_list, *min_node = NULL;
    for (int i = 0; i < n; ++i)
    {
        Node *q = head->next;
        min_node = q;
        while(q!=NULL)
        {
            if(min_node->Key<q->Key)
                min_node = q;
            q = q->next;
        }
        p->next = min_node;
        p = p->next;
    }
    head->next = sorted_list;
}
```

6-9

```
void bubble_sort(char A[][31], int n)//起泡排序
```

```
{
    int flag = 1;//排序好的标记
    for (int i = 0; i < n; ++i)
    {
        for (int j = 0; j < n - i - 1; ++j)
        {
            if (strcmp(A[j],A[j+1])>0)//两两比较，左比右大
            {
                flag = 0;//进行了排序
                swap(A[j], A[j + 1]);//交换
            }
        }
        if(flag)//如果没有经过交换则直接退出
            break;
    }
}
```

6-10

```
#include <iostream>
using namespace std;
typedef struct RNode{
    int key;
    int num;
    struct RNode *next;
}RNode,*RLink;
typedef struct{
    RLink head;
    RLink tail;
}List;
int main()
{
    int n=1000;//1000个数据
    List L[10];//十条链
    RLink p=new RNode;
    p->next=NULL;
    for(int i=0;i<=9;++i)
    {
        L[i].head=NULL;
        L[i].tail=NULL;
    }
    for(int i=1;i<=n;++i)
    {
        int t;
        cin>>t;//输入数据
        RLink q=new RNode;//建立新节点
        q->key=t;
        q->num=t;
        q->next=p->next;
        p->next=q;//前插法
    }
    RLink p1=p->next;
    for(int i=1;i<=3;i++)//分别按个位到百位排序
    {
        while(p1)//
        {
            RLink q=p1;
```

```
            RLink t=new RNode;
            t->next=NULL;
            t->key=q->key;
            int left=q->num%10;//当前的位
            t->num=q->num/10;
            if(L[left].head==NULL)
                L[left].head=t;
            else
                L[left].tail->next=t;
            L[left].tail=t;
            p1=p1->next;
        }
        int j=0;
        while(L[j].head==NULL)//找第一个头不为空的链
            j++;
        p1=L[j].head;
        for(int k=j+1;k<=9;k++)//进行链接
        {
            if(L[k].head==NULL)
                continue;
            L[j].tail->next=L[k].head;
            j=k;
        }
        for(int i=0;i<=9;i++)
        {
            L[i].head=NULL;
            L[i].tail=NULL;
        }
    }
    while(p1)//最后都在链上
    {
        cout<<p1->key<<endl;
        p1=p1->next;
    }
    return 0;
}
```

6-11 A

6-12 B

6-13

(1) 426 87 275 61 170 503 897 908 653 512

(2) 61 87 170 275 426 503 512 653 897 908

6-14

```
#include <climits>//INT_MAX
#include <iostream>
#include <vector>
using namespace std;
void treeSelectSort(int a[], int n)
{
    vector<int> tree(2 * n - 1, 0);//初始化树为完全二叉树
    for (int i = 0; i < n; ++i)
```

```
    {
        tree[n - 1 + i] = a[i];//将待排序元素置为叶结点
    }
    for (int i = 0; i < n; ++i)//循环n次找最小值
    {

        for (int j = 2 * n - 2; j > 0; j -= 2)//从后向前进行锦标赛
        {
            tree[(j - 1) / 2] = tree[j - 1] < tree[j] ? tree[j - 1] : tree[j];//
将较小值放在上一层
        }
        a[i] = tree[0];//取出当前的最小值
        for (int j = n - 1; j < 2 * n - 1; ++j)
        {
            if (tree[j] == a[i])
            {
                tree[j] = INT_MAX;//将最小值所在的位置设为正无穷，用来寻找下一个最小值
                break;
            }
        }
    }
}
int main()
{
    int a[]={49,38,65,97,76,13,27,49};
    int n = sizeof(a) / sizeof(a[0]);
    treeSelectSort(a,n);
    for(int i=0;i<n;++i)
    {
        cout << a[i] << ' ';
    }
    return 0;
}
```