

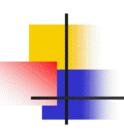
实验七

快速排序实验



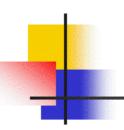
一、实验目的

- 掌握快速排序算法的基本思想
- 掌握快速排序的实现方法
- 掌握快速排序的时间性能



二、实验要求

- 熟悉C++语言编程
- ■掌握快速排序的原理



三、实验内容

1、问题描述

用快速排序实现对无序序列的排序。



三、实验内容

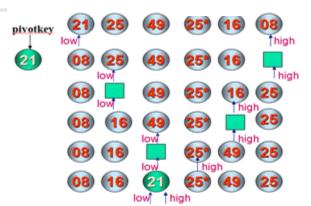
2、算法

基本思想:

- 任取待排序记录序列中的某个记录(例如取第一个记录)作为基准(枢),按照该记录的关键字大小,将整个记录序列划分为左右两个子序列:
- 左侧子序列中所有记录的关键字都小于或等于基准记录的关键字
- 右侧子序列中所有记录的关键字都大于基准记录 的关键字



三、实验内容 4000



2、算法

- 1、取序列第一个记录为枢轴记录,其关键字为Pivotkey;指针 low指向序列第一个记录位置(low=1),指针high指向序列 最后一个记录位置(High=SeqList.Len)
- 2、从high指向的记录开始,向前找到第一个关键字的值小于 Pivotkey的记录,将其放到low指向的位置, low++
- 3、从low指向的记录开始,向后找到第一个关键字的值大于 Pivotkey的记录,将其放到high指向的位置,high--
- 4、重复2、3,直到low==high,将枢轴记录放在low(high)指 向的位置



三、实验内容

3、输入

共一行:第一个数字n表示样本数目,其后跟 n个样本



三、实验内容

4、输入样本

8 5 6 7 9 3 4 8 2



三、实验内容

5、输出

■ 第一行:原始样本序列

■ 第二行: 第一趟快速排序结果

■ 第三行: 最终排序结果



三、实验内容

6、输出样本

5 6 7 9 3 4 8 2

2 4 3 5 9 7 8 6

2 3 4 5 6 7 8 9



四、实验步骤

- 1、顺序表的定义
- 2、快速排序函数
- 3、顺序表显示函数
- 4、主函数



三、实验内容

1、顺序表的定义

■ 定义顺序表的结构、表长:

```
#define MAXLISTLEN 20
struct List {
    int Key[MAXLISTLEN];
    int Len;
};
List SeqList;
```



三、实验内容

2、快速排序函数

```
int FirstQuick = 'T':
void QuickSort(int low, int high)
{ int i, j, Pivotkey;
  i = low; j = high;
                                 //记录顺序表的上、下界
  Pivotkey = SeqList. Key[low];
                                 //当high>low的时候循环
  while(low <high) {
  SeqList. Key[low] = Pivotkey;
                                               // low = high
```



三、实验内容

2、快速排序函数

```
·
//显示第一趟快速排序结果
if (FirstQuick=='T') ShowSeqList();
FirstQuick = 'F';

//对子序列数组进行递归快速排序
if (i<low-1) QuickSort(i, low-1);
if (high+1<j)QuickSort(high+1, j);
```

```
5 6 7 9 3 4 8 2
2 4 3 5 9 7 8 6
2 3 4 5 6 7 8 9
```

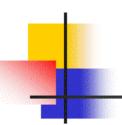


三、实验内容

3、顺序表显示函数

显示顺序表

```
void ShowSeqList()
{
   int i;
   for (i=1; i<SeqList.Len; i++) cout << SeqList.Key[i] << " ";
   cout << SeqList.Key[i] << endl;
   // 最后一个元素与回车之前不留空格
}
```



三、实验内容

4、主函数

```
int main(int argc, char *argv□)
  int i:
  cin >> SeqList. Len;
  for (i=1; i \le SeqList.Len; i++)
                                        // 输入顺序表
      cin >> SeqList. Key[i];
                                        // 显示原始序列
  ShowSeqList();
                                        // 快速排序
  QuickSort(1, SeqList.Len);
                                        // 显示最终排序结果
  ShowSeqList();
  return 1:
```