# Comprehensive Experiment 3 :

# Sort comprehensive experiment

Class: 计科20-1 Experiment Date: 2021/12/6

Student ID 1: 20402080240 Name 1: 王夏禹

Student ID 2：20402070113 Name 2：瞿子椋

Student ID3: 20401010121 Name 3: 徐志伟

**One. Experimental purpose**

1) Familiar with the basic operations of the sort.

2) Master the operation of various internal sorting.

3) Deepen the understanding of the sort, and to develop the programming ability of solving practical problems gradually.

**Two. Experimental environment**

Computers equipped with Visual C6.0/CFree.

The experiment lasted for 4 hours.

**Three. Experimental content**

A series of strings are stored in a two-dimensional array. Try to sort them with some sorting algorithms (at least two algorithms, such as insert sorting, bubble sorting, quick sorting, and heap sorting). You should sort them to dictionary order finally.

For example: two-dimensional array is :

char s[][20]={“while”，”if”，“else”，”do”，“for”，”switch”，“case”};

**Four. Important data structures**

void InsertSort(char R[][20],int n)

{

int i,j;//i是进行推进，j负责i之前的

char temp[1][20];

for(i=1;i<n;i++)

{

for(int p=0;p<20;p++)//temp为标量

{

temp[0][p]=R[i][p];

}

j=i-1;//j表示的位置是标量前一位的位置

while(j>=0&&(temp[0][0]<R[j][0]))//标量小于前一位数

{

for(int p=0;p<20;p++)//标量和前一位数交换位置

{

R[j+1][p]=R[j][p];

}

--j;//标量前移，继续比较

}

for(int p=0;p<20;p++)//最后停止的位置赋值为标量

{

R[j+1][p]=temp[0][p];

}

}

}

void SelectSort(char R[][20],int n)

{

int i,j,k;

char temp[1][20];

for(i=0;i<n;i++)

{

k=i;//从0开始，每次循环找到一个最小数，遍历范围缩小

for(j=i+1;j<n;j++)

{

if(R[k][0]>R[j][0])//j是遍历指针，k是j之后的标量指针负责找到最小的数

{

k=j;

}

}

//让最小的数与序列的第一个关键字换一下位置

for(int p=0;p<20;p++)

{

temp[0][p]=R[i][p];

}

for(int p=0;p<20;p++)

{

R[i][p]=R[k][p];

}

for(int p=0;p<20;p++)

{

R[k][p]=temp[0][p];

}

}

}

void QuickSort(char R[][20],int low,int high)

//sorted from R[low] to R[high]

{

char temp[1][20];

int i=low,j=high;

if(low<high)

{

for(int p=0;p<20;p++)//temp=first key in the array

{

temp[0][p]=R[low][p];

}

while(i<j)

{

//j从队尾开始向前遍历

while(j>i&&R[j][0]>=temp[0][0])//j从右往左扫描，找到一个小于temp的关键字

{

j--;

}

if(i<j)

{

for(int p=0;p<20;p++)//i的位置放入j的数

{

R[i][p]=R[j][p];

}

i++;//i向后移一位

}

//i开始从队前向后遍历

while(i<j&&R[i][0]<temp[0][0])//i从左向右扫描，找到一个大于temp的关键字

{

i++;

}

if(i<j)

{

for(int p=0;p<20;p++)//j的位置放入i的数

{

R[j][p]=R[i][p];

}

j--;//j左移一位

}

}

//此时i=j

for(int p=0;p<20;p++)//在中轴位置填入temp

{

R[i][p]=temp[0][p];

}

QuickSort(R, low, i-1);//递归地对temp左边的关键字进行排序

QuickSort(R, i+1, high);//递归地对temp右边的关键字进行排序

}

}

**Five. Realization idea analysis**

首先，选择文件储存句子的方式，然后选择了直接插入排序、选择排序和快速排序进行句子里的单词排序。

**Six. Program debugging problem analysis**

第一次遇到的bug就是，无法给二维数组赋值，但是后来采用了一个个字母赋值，问题得到解决。第二次是发现句子里The 和Dogge排序出现错误，后来发现大写和小写的ascii码不一样导致的。

**Seven. Experimental summary**

首先，我又掌握了从文件中读取单词用二维数组储存。

其次，我学习了直接插入排序、选择插入排序和快速排序。

最后，了解了如何给二维数组赋值，让二维数组作为行参

**Eight. Crew Division**

|  |  |  |
| --- | --- | --- |
| **Group division** | | |
| **Member name** | **Work done** | **Completion situation** |
| **王夏禹** | **代码设计** | **已完成** |
| **瞿子椋** | **实验报告** | **已完成** |
| **徐志伟** | **任务分配** | **已完成** |