1. 问题描述与分析

1.已知某公司设计部人员如表，请选用一种存储结构将其进行存储，并选用一种排序算法按照工号为关键字进行递增排序。

2.抽象数据类型定义

typedef struct {

int number; //工号

String sex; //性别

String name; //姓名

} node; //存储员工信息的结构体

3输入值范围

按题目要求设计工号为整形，性别和姓名设为string类型，

问题解决思路

要想存储多个员工信息，考虑到那几种数据结构的用途为了方便增删改查，选用表（数组）结构。用到的是结构体数组。数组的每个元素都是单独的结构体。结构体中存放着员工的基本信息。这个基本信息在以后的过程还可以不断增加，比如电话号码，家庭住址等。

以题目的排序要求，我们肯定要把每个员工的工号抽出来排序。我想到的有两种思路，一是排序好后在原来空间上直接交换，不过这样时间效率有点低，所以干脆以空间换取时间，再创一个结构体数组，排序好后一一填入。

这里的排序用到简单选择排序。

1. 算法及算法分析
2. 结构体。

#include<iostream>

#include<string>

using namespace std;

typedef struct {

int number; //工号

string sex; //性别

string name; //姓名

} node; //存储员工信息的结构体

void selectSort(int a[], int n) { //简单选择排序

int min, temp;

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

min = i;

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

if (a[j] < a[min])

min = j;

}

if (i != min) {

temp = a[i];

a[i] = a[min];

a[min] = temp;

}

}

}

int main() {

int i;

node employee[8] = {

{23,"男","王火"},

{12,"男","宜丰"},

{45,"女","张小花"},

{18,"男","李超"},

{19,"女","万思思"},

{77,"女","吴方"},

{4,"男","刘一明"},

{96,"男","邓伟"}

};

cout<<"员工信息如下"<<endl;

cout << "工号\t性别\t姓名" << endl;

for (i = 0; i < 8; i++) {

cout << employee[i].number << "\t" << employee[i].sex << "\t" << employee[i].name << endl;

}

int traser[8];

for (i = 0; i < 8; i++) {

traser[i] = employee[i].number;

}

selectSort(traser, 7);

node employee2[8];

int j = 0;

int k = 0;

for (i = 0; i < 8; i++) {

while (traser[j] != employee[k].number) {

k++;

}

employee2[i].number = employee[k].number;

employee2[i].sex = employee[k].sex;

employee2[i].name = employee[k].name;

j++;

k = 0;

}

cout<<"按工号排序后信息如下："<<endl;

cout << "工号\t性别\t姓名" << endl;

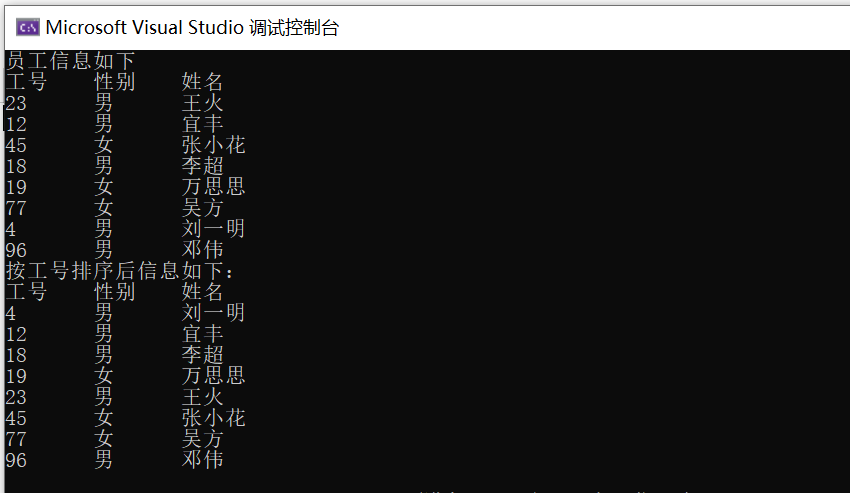
for (i = 0; i < 8; i++) {

cout << employee2[i].number << "\t" << employee2[i].sex << "\t" << employee2[i].name << endl;

}

return 0;

}



算法性能分析：

信息的输入和输出时间复杂度都为O（n) 空间复杂度为O（1）

简单选择排序算法： 时间复杂度都为O（n^2) 空间复杂度为O（1）