

**数据结构课程设计**

**题目：** 用堆实现优先队列

**专业班级：** 计算机科学与技术 6班

**学 号：** 20204331

**姓 名：** 王博文

**指导教师：** 隋毅

**报告日期：** 2022年3月20日

计算机科学技术学院

**2022年3月30日**

一、课程设计内容

按照菜单内容实现优先队列的一些基本操作。

说明：使用顺序存储结构实现。

**菜单内容：**

**1. 优先队列的插入操作**

**2. 最大优先级元素出队列操作**

**3. 调整优先队列某个元素的优先级**

**4. 求优先队列的长度**

**5. 优先队列的查询操作**

**6. 退出系统**

**请选择：1 – 6：**

二、算法设计

**本算法我运用了小根堆作为堆排序的依据，每次堆排序后都为将最后一个数据变为权值最大的数据，在插入元素之后我会对其进行一次排序并将其打印出来，每次更改一个元素的时候，我也会对其进行排序。总的来说这个实验主要就是考验堆排序的熟练程度，如果堆排序能顺利地写出来，那么其他的功能也就迎刃而解了。**

三、核心代码实现

#include<iostream>

using namespace std;

typedef struct Queue {

int num;

int weight;

} Queue;

int n=0;

void Insertqueue(Queue a[20]) {

cout<<"请输入想要插入的数据及优先级，输入0即停止："<<endl;

int x,y;

while(1) {

cin>>x>>y;

if(x==0||y==0) {

break;

} else {

a[n].num=x;

a[n].weight=y;

n++;

}

}

n--;

}

void swap(Queue \*q,Queue \*p) {

Queue t;

t=\*q;

\*q=\*p;

\*p=t;

}

void Printqueue(Queue a[20]) {

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

cout<<a[i].num<<" ";

}

cout<<endl;

}

void Heapsort(Queue a[20]) {

int i=n;

int max;

int m=0;

int k=n;

while(m!=n) {

if(i%2==0) {

max=i;

if(a[i].weight<a[i-1].weight)

max=i-1;

else

max=i;

if(a[i/2-1].weight<a[max].weight)

swap(&a[i/2-1],&a[max]);

i=i-2;

} else {

max=i;

if(a[i/2].weight<a[max].weight)

swap(&a[i/2],&a[max]);

i=i-1;

}

for(i; i>0; i=i-2) {

if(a[i].weight<a[i-1].weight)

max=i-1;

else

max=i;

if(a[i/2-1].weight<a[max].weight)

swap(&a[i/2-1],&a[max]);

}

swap(&a[k],&a[0]);

k--;

i=k;

m++;

}

}

void popqueue(Queue a[20]) {

cout<<"值："<<a[n].num<<" "<<"权值："<<a[n].weight<<endl;

n--;

}

void correctqueue(Queue a[20]) {

cout<<"请输入想要调整元素的优先级："<<endl;

int x,y;

cin>>x>>y;

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

if(a[i].num==x) {

a[i].weight=y;

}

}

Heapsort(a);

}

void sumqueue(Queue[20]){

cout<<"优先队列长度为："<<n+1<<endl;

}

void searchqueue(Queue a[20]){

cout<<"请输入想要查询的值"<<endl;

int q;

cin>>q;

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

if(a[i].num==q) {

cout<<"值："<<a[i].num<<"权值："<<a[i].weight<<endl;

}

}

}

int main() {

cout<<"1.优先队列的插入操作"<<endl<<"2.最大优先级元素出队列操作"<<endl<<"3.调整优先队列某个元素的优先级"<<endl;

cout<<"4.求优先队列的长度"<<endl<<"5.优先队列的查询操作"<<endl<<"6.退出系统"<<endl;

Queue a[20];

while(1) {

int z;

cin>>z;

switch(z) {

case 1:

Insertqueue(a);

Heapsort(a);

Printqueue(a);

break;

case 2:

popqueue(a);

Printqueue(a);

break;

case 3:

correctqueue(a);

Heapsort(a);

Printqueue(a);

break;

case 4:

sumqueue(a);

break;

case 5:

searchqueue(a);

break;

case 6:

exit(0);

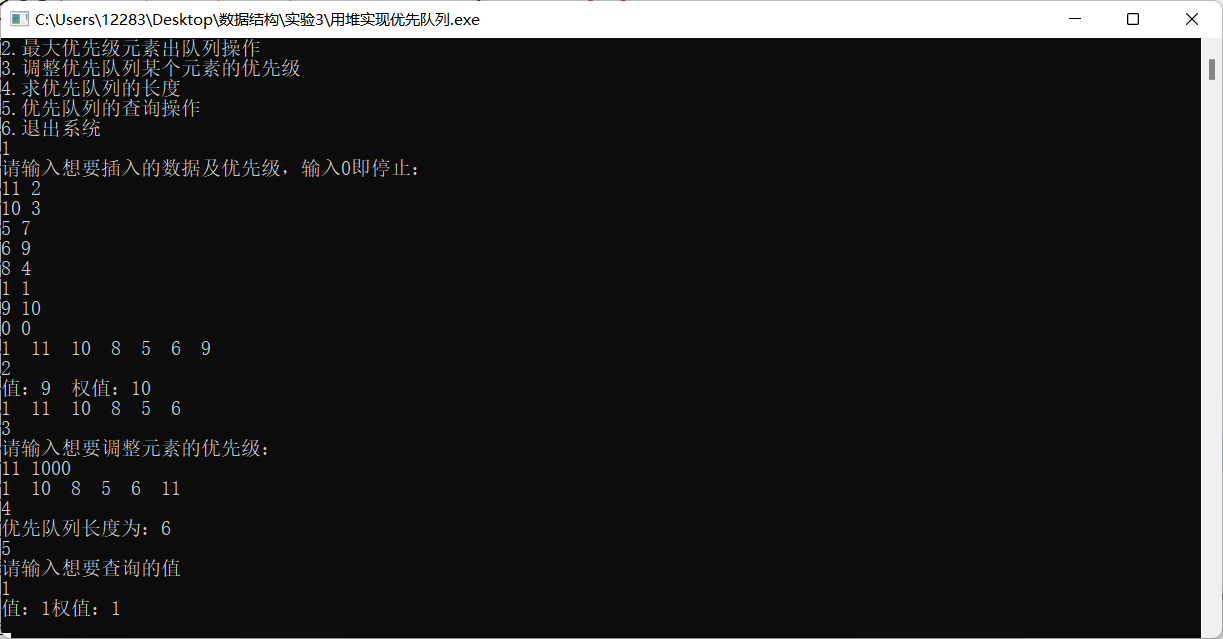
}

}

return 0;

}

四、测试与分析



五、总结及体会

通过这次的实验巩固了我对堆排序的认识，上个学期只是浅浅的了解了一下原理，并没有真正的将其落实，经由这次写代码的经历丰富了我对结构体之间互换的理解，传的时候要传输他的地址，而swap的时候要换他们的本体。

1. 参考书目和网络资源

无