**C++ 语 言 程 序 设 计**

实

验

报

告

实 验 一

姓名： 刘健恒

学号： SZ170320207

班级： 自动化2班

**一 实验项目**

1. 熟悉C++程序设计
2. 基于模板，编写堆排序算法

**二 实验原理**

1. 说明堆排序实现的原理与操作方法（给出算法的流程图与简要说明）

原理：按照父节点大于子节点的原则建立堆，则顶点数值是最大值。然后依次将堆的最大值置于末尾，待循环完成完成排序。

1. 算法的程序实现与结果（给出算法的源代码，说明关键代码的操作含义，给出运行结果）

#include <iostream>

#include <string>

using namespace std;

template <typename T>

void swap1(T &a, T &b)

{

T s;

s = a;

a = b;

b = s;

}

template <typename T>

void insert\_i(T a[],int length)

{

if ((a[length])>(a[(length-1)/2]))

{

swap1(a[length],a[(length-1)/2]);

insert\_i(a,(length-1)/2 );

}

}

template <typename T>

void delete\_max (T a[], int i, int length)

{

int n=2\*i+2;

if(n<length)

{

if(a[n]>a[n-1])

{

a[i]=a[n];

delete\_max(a,n,length);

}

else

{

a[i]=a[n-1];

delete\_max(a,n-1,length);

}

}

else

{

a[i]=a[length];

insert\_i(a,i);

}

}

template <typename T>

void heapSort (T a[],int n)

{

cout<<"========================="<<endl;

cout<<"通过插入过程实现建立堆"<<endl;

cout<<"========================="<<endl;

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

{

insert\_i(a,i);

cout<<i<<" : ";

for (int t = 0; t<n; t++)

{

cout<<" "<<a[t]<<" ";

}

cout<<endl;

}

cout<<"========================="<<endl;

cout<<"通过删除过程实现排序"<<endl;

cout<<"========================="<<endl;

for (int i = n-1; i>=0; i--)

{

T s=a[0];

delete\_max(a,0,i);

a[i]=s;

cout<<5-i<<" : ";

for (int t = 0; t<n; t++)

{

cout<<" "<<a[t]<<" ";

}

cout<<endl;

}

cout<<"========================="<<endl;

cout<<"排序结果为：";

for (int t = 0; t<n; t++)

{

cout<<" "<<a[t]<<" ";

}

cout<<endl;

cout<<"========================="<<endl;

}

int main()

{

int n = 6;

int a[6]= {1,4,5,3,2,8};

double d[6]= {1.4,4.3,2.5,3.7,9.2,0.8};

char c[6]= {'c','d','r','a','y','b'};

string s[6]= {"hello","yes","byebye","welcome","good","morning"};

heapSort(a,n);

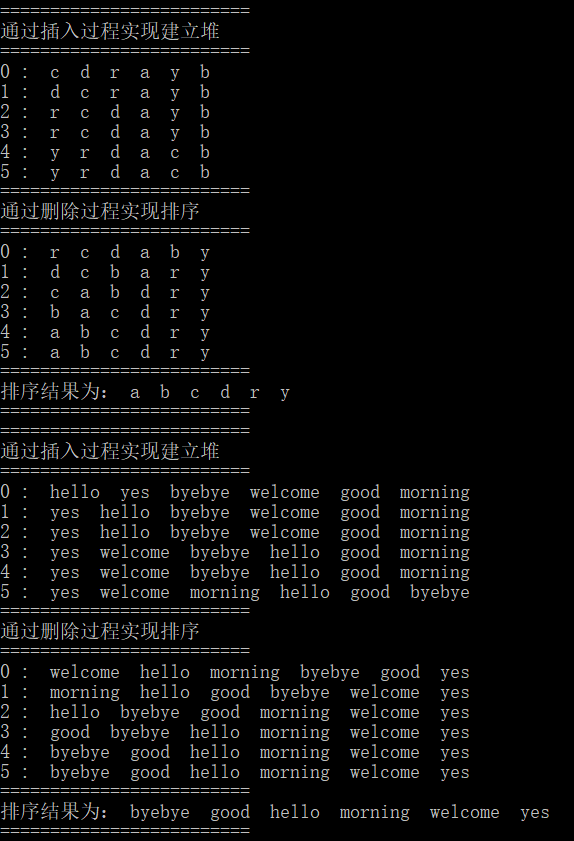
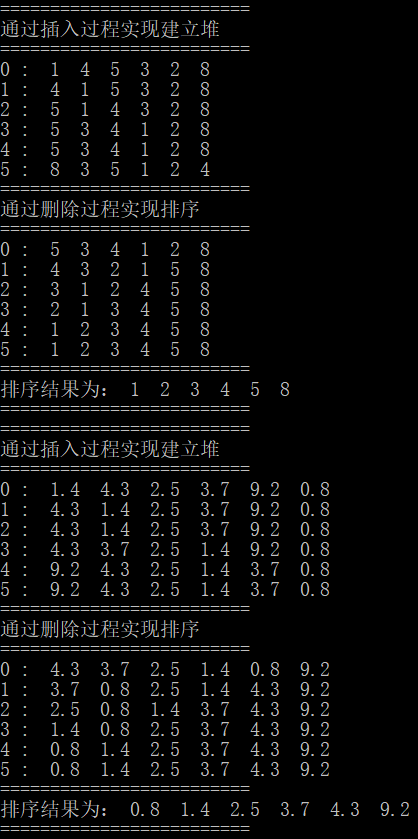
heapSort(d,n);

heapSort(c,n);

heapSort(s,n);

return 0;

}



**三 实验总结与建议**

（总结实验实施过程，说明实验过程中遇到的问题与解决方案；提出实验环节的建议）

一开始对堆排序逻辑不清楚，进行简单的画图使得逻辑梳理清楚。编写代码时需要尝试理解示例代码函数试图实现的功能，较难。自行编写可能更快些。

本次实验使得我更加熟悉C++的操作以及学习了堆排序节省数据处理时间。

**四 排序效率评估**

