# **堆相关笔记**

​

参考笔记：柳婼大佬的总结https://www.liuchuo.net/archives/2277

​

### **一、二次总结理由**

关于二次总结是否有必要，我觉得是有的，参考学习别人的知识，内化的过程是一次总结的过程。每次参考同一个人的思维轨迹，确实能培养思维。该篇笔记整理理由：在PAT Advanced level的Heap Path一题，参考了关于堆的正序遍历的镜像。于是参考一下堆相关笔记。

### **二、堆数据结构主要算法(大根堆为例)**

创建堆

void createHeap(){  
 for(int i=n/2;i>=1;i--)  
 downAjust(i,n);  
}

向下调整：思想是，low代表孩子，high代表极限。我们在[low,high]范围进行调整，low和low+1为2\*low的孩子。如果没有交换，那就break掉，交换了还要和上面父亲去比较。

void downAdjust(int low,int high){  
 int i=low,j=i\*2;//i为要调整的节点，j为左孩子  
 while(j<=high){  
 if(j+1<=high && heap[j+1]>heap[j]) j=j+1;  
 if(heap[j]>heap[i]){  
 swap(heap[j],heap[i]);  
 i=j;j=i\*2;  
 }else break;  
 }  
}

删除堆顶元素

void deleteTop(){  
 heap[1]=heap[n--];//用第n个数进行覆盖  
 downAdjust(1,n);//之后进行向下调整第一个数  
}

增加一个元素

void insert(int x){  
 heap[++n]=x;  
 upAdjust(1,n);  
}

向上调整

void upAdjust(int low,int high){  
 int i=high,j=i/2;  
 while(j>=low){  
 if(heap[j]<heap[i]){  
 swap(heap[j],heap[i]);  
 i=j;j=i/2;  
 }else break;  
 }  
}

堆排序

将第一个数和顶部元素互换，进行向下调整（1，i-1）的范围，直到i==2

void heapSort(){  
 createHeap();  
 for(int i=n;i>=2;i--){  
 swap(heap[i],heap[1]);  
 downAdjust(1,i-1);  
 }  
}

堆排序的时间复杂度：O(nlogn) 空间复杂度为 O(1) 稳定性为不稳定

### **三、手写heap.h头文件**

#ifndef HEAP\_H\_INCLUDED  
#define HEAP\_H\_INCLUDED  
​  
#define maxn 1000  
​  
using namespace std;//这里面有swap函数  
​  
int heap[maxn];  
​  
void upAdjust(int low,int high){  
 int i=high,j=i/2;  
 while(j>=low){  
 if(heap[j]<heap[i]){  
 swap(heap[j],heap[i]);  
 i=j;j=i/2;  
 }else break;  
 }  
}  
​  
void downAdjust(int low,int high){  
 int i=low,j=i\*2;//i为要调整的节点，j为左孩子  
 while(j<=high){  
 if(j+1<=high && heap[j+1]>heap[j]) j=j+1;  
 if(heap[j]>heap[i]){  
 swap(heap[j],heap[i]);  
 i=j;j=i\*2;  
 }else break;  
 }  
}  
​  
void createHeap(int n){  
 for(int i=n/2;i>=1;i--)  
 downAdjust(i,n);  
}  
​  
​  
void deleteTop(int n){  
 heap[1]=heap[n--];//用第n个数进行覆盖  
 downAdjust(1,n);//之后进行向下调整第一个数  
}  
​  
void insert(int x,int n){  
 heap[++n]=x;  
 upAdjust(1,n);  
}  
​  
​  
void heapSort(int n){  
 createHeap(n);  
 for(int i=n;i>=2;i--){  
 swap(heap[i],heap[1]);  
 downAdjust(1,i-1);  
 }  
}  
#endif // HEAP\_H\_INCLUDED

### **四、测试文件**

#include <iostream>  
#include "heap.h"  
using namespace std;  
int main(){  
   extern int heap[maxn];  
   extern int n;  
   scanf("%d",&n);  
   /\*\*我们的堆序列是从[1,n]的\*/  
   for(int i=1;i<=n;i++) scanf("%d",&heap[i]);  
   createHeap();//生成大根堆  
   insert(4);//插入堆顶元素  
   deleteTop();//删除堆顶  
   heapSort();//排序  
   for(int i=1;i<=n;i++) printf("%d%s",heap[i],i==n?"\n":" ");  
   system("pause");  
   return 0;  
}