堆就是一种特殊的完全二叉树，特殊在哪里呢？就是父节点的值（注意不是值）小于子节点的值。

又因为有这样的一种规律，父节点的编号 \* 2等于父节点的左子节点的编号。当然加上1就是父节点的右子节点的编号。

这样的话给你一个父节点，你就能很轻松访问其子节点。

那么怎么创建一个最小堆呢？

先用一个数组a来存储所有的节点。a的下标就是二叉树的编号。一开始二叉树是杂乱无章的、乱序的。

先说下建立最小堆的思路。因为最小堆是父节点的值小于子节点的值，所以我们只要遍历我们要建立的堆中的有子节点的节点，让它的符合最小堆，那么这个堆也就是最小堆了（贪心策略？）。

在开始之前我们要明白一个概念，就是叶节点。就是该节点如果没有子节点就是叶节点。也就是说我们根本不用遍历叶节点，因为他们本身就符合最小堆的要求。所以我们建立最小堆时遍历每个点时可以忽略叶节点。而叶节点刚好占全部节点的一半，就是说a数组的后一半都是叶节点。（如果总结点为奇数，那就是叶节点要比不是叶节点的多1）

//已输入数据到a数组，n为节点总数

for(int i = n / 2;i >= 1;i--)//开始遍历每个非叶子结点

{

//判断每个节点是否符合最小堆要求

}

怎么调整为最小堆呢？

我们从最大的非叶子结点开始判断，如果当前结点的父子节点的值小于子节点的值，找到最小的子节点，跟父节点交换位置。一直遍历到根节点，就可以得到最小堆了。

for(int i = n / 2;i >= 1;i--)//开始遍历每个非叶子结点

{

flag = 0;

//判断每个节点是否符合最小堆要求

While(flag == 0)//flag == 1时退出循环

{

If(a[i] > a[i \* 2])

{

temp = i \* 2;

}

Else

{

temp = i;

}

If(h[temp] > h[i \* 2 + 1])

temp = i \* 2 + 1;

If(temp != i)

{

tmp = a[temp];

a[temp] = a[i];

a[i] = tmp;

}

flag = 1;

}

}

这样输出的话输出第一个就好了，然后把堆的最后一个点赋值到堆顶。在建立一次最小堆就好了。

Num = n;

for(int i = 1;i <= Num;i++)

{

//输出堆顶

a[1] = a[n];

n--;

//在进行一次建立堆的过程

}

其实建立堆的过程相当于让调整父节点和子节点的位置。选取的节点只要关心父节点和子节点的值就好了。