5.堆

堆（是一个完全二叉树：除最后一层，都是满的，最后一层从左向右排列）

eg: 小根堆，每一个点小于左右儿子

- 用一维数组存储， 左儿子2x, 右儿子2x+1
- 下标从1开始

操作	代码
插入一个数	heap[++size] = x **up(size)**
求集合中的最小值	heap[1]
删除最小值	heap[1] = heap[size]; size--; **down(1)**
删除任意一个元素	heap[k] = heap[size]; size--; down[k]; up[k]
修改任意一个元素	heap[k] = x; down[k]; up[k]

```
int h[N], size;

//down
//把一个值变大时候down
//跟左右节点的最小值交换
void down(int u)
{
    int t = u;
    if (u * 2 <= size && h[u * 2] < h[t]) t = u * 2;
    if (u * 2 + 1 <= size && h[u * 2 + 1] < h[t]) t = u * 2 + 1;
    if (u != t)  //最小值的下标不同的话
    {
        swap(h[u], h[t]);
        down(t);
    }
}
//up
//把一个值变小时候up
//跟父节点交换，因为小根堆
void up(int u)
{
    while (u / 2 && h[u] < h[u / 2])
    {
        swap(h[u], h[u / 2]);
        u >>= 1;
    }
}

// O(n)建堆
```

```
30 for (int i = n / 2; i; i -- ) down(i);
```

STL

- 优先队列具有队列的所有特性，包括队列的基本操作，只是在这基础上添加了内部的一个排序，它本质是一个堆实现的。
- 定义priorith_queue<类型, vector<类型>, cmp>;

升序队列，小顶堆

- priority_queue <int,vector<int>,greater<int> > q;

降序队列，大顶堆

- priority_queue <int,vector<int>,less<int> >q;