堆的完全二叉树实现代码

主讲人：邓哲也

堆的完全二叉树实现代码

- 首先定义一个数组h[]储存堆元素。
- heap_size表示堆的大小。

- int h[N], heap_size;

堆的完全二叉树实现代码

- top操作，只需返回根结点。

- int top() {
- return h[1];
- }

堆的完全二叉树实现代码

- shift_up操作，从节点x开始向根方向依次比较。
- 注意x的父节点是x/2。

```
void shift_up(int x) {
    if (x == 1) return;
    if (h[x] < h[x / 2]) {
        swap(h[x], h[x / 2]);
        shift_up(x / 2);
    }
}
```

堆的完全二叉树实现代码

- shift_down操作，从节点x开始向子节点方向比较。
- 注意x的左子节点是2x，右子节点是2x+1。
- 下标不要超过heap_size。

```
void shift_down(int x) {
    if (x * 2 > heap_size) return;
    int k = x;
    if (h[x * 2] < h[k]) k = x * 2;
    if (x * 2 + 1 <= heap_size && h[x * 2 + 1] < h[k]) k = x * 2 + 1;
    if (k == x) return;
    swap(h[x], h[k]);
    shift_down(k);
}
```

堆的完全二叉树实现代码

- pop操作，先把要返回的h[1]存下来，然后和最后一个节点交换，向下比较

```
int pop() {
    int val = h[1];
    h[1] = h[heap_size];
    heap_size --;
    shift_down(1);
}
```

堆的完全二叉树实现代码

- push操作，插在最后一个节点，然后向上比较。

```
void push(int v) {
    h[++ heap_size] = v;
    shift_up(heap_size);
}
```

下节课再见

zxwanmen.org