

小白专场：堆中的路径

题意理解

将一系列给定数字插入一个初始为空的小顶堆 $H[]$ 。随后对任意给定的下标 i ，打印从 $H[i]$ 到根结点的路径。

输入样例:

5 3 5: 5个结点 3: 进行三次查询

46 23 26 24 10

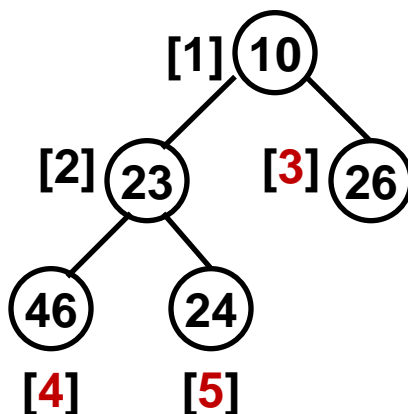
5 4 3

输出样例:

24 23 10

46 23 10

26 10



堆的表示及其操作

堆是按照一定顺序组织的完全二叉树

```
#define MAXN 1001
#define MINH -10001
```

```
int H[MAXN], size;
```

size表示当前堆的大小

```
void Create ()
```

```
{
```

```
    size = 0;
```

```
    H[0] = MINH;
```

```
    /*设置“岗哨”*/
```

```
}
```

```
void Insert ( int X )
```

```
{
```

```
    /* 将X插入H。这里省略检查堆是否已满的代码 */
```

```
    int i;
```

```
    for (i=++size; H[i/2] > X; i/=2)
```

```
        H[i] = H[i/2];
```

```
    H[i] = X;
```

```
}
```

主程序

```
int main()
{
    // n:代表这个堆的元素个数
    // m:代表我要查询的次数
    ●读入n和m
    ●根据输入序列建堆
    ●对m个要求: 打印到根的路径

    return 0;
}
```

```
int main()
{
    int n, m, x, i, j;

    scanf("%d %d", &n, &m);
    Create(); /* 堆初始化 */
    for (i=0; i<n; i++) { /* 以逐个插入方式建堆 */
        scanf("%d", &x);
        Insert(x);
    }
    for (i=0; i<m; i++) {
        scanf("%d", &j);
        printf("%d", H[j]);
        while (j>1) { /* 沿根方向输出各结点 */
            j /= 2;
            printf(" %d", H[j]);
        }
        printf("\n");
    }

    return 0;
}
```