

堆排序

题目

提交记录

讨论

题解

视频讲解

输入一个长度为 n 的整数数列，从小到大输出前 m 小的数。

输入格式

第一行包含整数 n 和 m 。

第二行包含 n 个整数，表示整数数列。

输出格式

共一行，包含 m 个整数，表示整数数列中前 m 小的数。

数据范围

$1 \leq m \leq n \leq 10^5$,

$1 \leq \text{数列中元素} \leq 10^9$

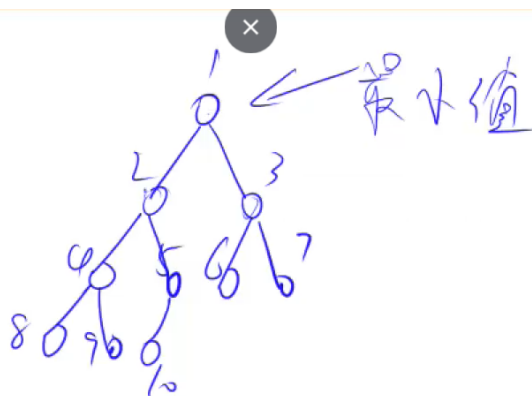
输入样例：

```
5 3
4 5 1 3 2
```

输出样例：

```
1 2 3
```

小根堆



存储

根节点



x 的左儿子: $2x$

x 的右儿子: $2x+1$

如何手写一个堆?

- | | |
|--------------|--|
| 1. 插入一个数 | <code>heap[++ size] = x; up(size);</code> |
| 2. 求集合当中的最小值 | <code>heap[1];</code> |
| 3. 删除最小值 | <code>heap[1] = heap[size]; size -- ; down(1);</code> |
| 4. 删除任意一个元素 | <code>heap[k] = heap[size]; size -- ; down(k); up(k);</code> |
| 5. 修改任意一个元素 | <code>heap[k] = x; down(k); up(k);</code> |

I

▲ 分析

35 i为什么从 $n/2$ 开始down?

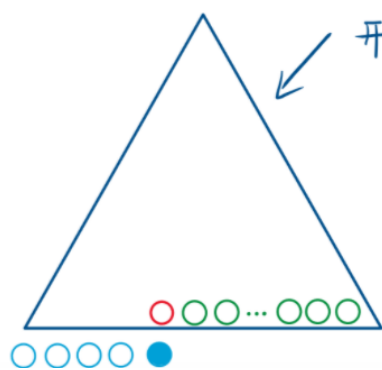
▼ 首先要明确要进行down操作时必须满足左儿子和右儿子已经是个堆。

★ 开始创建堆的时候, 元素是随机插入的, 所以不能从根节点开始down, 而是要找到满足下面三个性质的结点:

15

1. 左右儿子满足堆的性质。
2. 下标最大 (因为要往上遍历)
3. 不是叶结点 (叶节点一定满足堆的性质)

那这个点为什么是 $n/2$? 看图。



开始时堆是这样的

图中蓝色结点与绿色结点均为叶子结点, 显然叶子结点满足堆的性质。●是○的叶结点, (也就是整个堆的最后一个结点, 下标为 n), 无非就两种情况: 要么是左儿子要么是右儿子, 他们的父亲都是 $\frac{n}{2}$, 因此结点○满足前面的三个性质也就是从该结点开始down操作, 即 i 从 $\frac{n}{2}$ 开始down。

y总代码:

```

1 #include <iostream>
2 #include <algorithm>
3
4 using namespace std;
5
6 const int N = 100010;
7
8 int n, m;
9 int h[N], size;
10
11 void down(int u)
12 {
13     int t = u;
14     if (u * 2 <= size && h[u * 2] < h[t]) t = u * 2;
15     if (u * 2 + 1 <= size && h[u * 2 + 1] < h[t]) t = u * 2 + 1;
16     if (u != t)
17     {
18         swap(h[u], h[t]);
19         down(t);
20     }
21 }
22
23 int main()
24 {
25     scanf("%d%d", &n, &m);
26     for (int i = 1; i <= n; i ++ ) scanf("%d", &h[i]);
27     size = n;
28
29     for (int i = n / 2; i; i -- ) down(i);
30
31     while (m -- )
32     {
33         printf("%d ", h[1]);
34         h[1] = h[size];
35         size -- ;
36         down(1);
37     }
38 }

```

```

#include <iostream>
#include <algorithm>

using namespace std;

const int N = 100010;

int n, m;
int h[N], cnt;

void down(int u)
{
    int t = u;
    if (u * 2 <= cnt && h[u * 2] < h[t]) t = u * 2;
    if (u * 2 + 1 <= cnt && h[u * 2 + 1] < h[t]) t = u * 2 + 1;
    if (u != t)
    {
        swap(h[u], h[t]);
        down(t);
    }
}

int main()
{
    scanf("%d%d", &n, &m);
    for (int i = 1; i <= n; i ++ ) scanf("%d", &h[i]);
    cnt = n;

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

    while (m -- )
    {
        printf("%d ", h[1]);
        h[1] = h[cnt -- ];
    }
}

```

```
        down(1);  
    }  
  
    puts("");  
  
    return 0;  
}
```

作者: yxc

链接: <https://www.acwing.com/activity/content/code/content/45296/>

来源: Acwing

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。