16 第五讲 作业记录（一）

笔记本: 浙江大学《数据结构》
创建时间: 2025/4/12 16:08 **更新时间:** 2025/4/12 16:17
作者: panhengye@163.com
URL: https://pintia.cn/problem-sets/1873565885118418944/exam/problems/type/7?...

05-树7 堆中的路径

提交结果

题目	用户	提交时间
05-树7	飞翔的小师弟	2025/04/12 16:07:06
编译器	内存	用时
Python (python3)	3156 / 65536 KB	28 / 400 ms
状态	分数	评测时间
答案正确	25 / 25	2025/04/12 16:07:06

评测详情

测试点	提示	内存(KB)	用时(ms)	结果	得分
0	sample 调整到根、到中间位置，有不需要调整的元素	3084	28	答案正确	13 / 13
1	路径更长，交错，index从中间开始，有负数	3152	22	答案正确	6 / 6
2	最小N和M	3156	20	答案正确	2 / 2
3	最大N和M随机，元素取到正负10000	3152	25	答案正确	4 / 4

【代码记录】

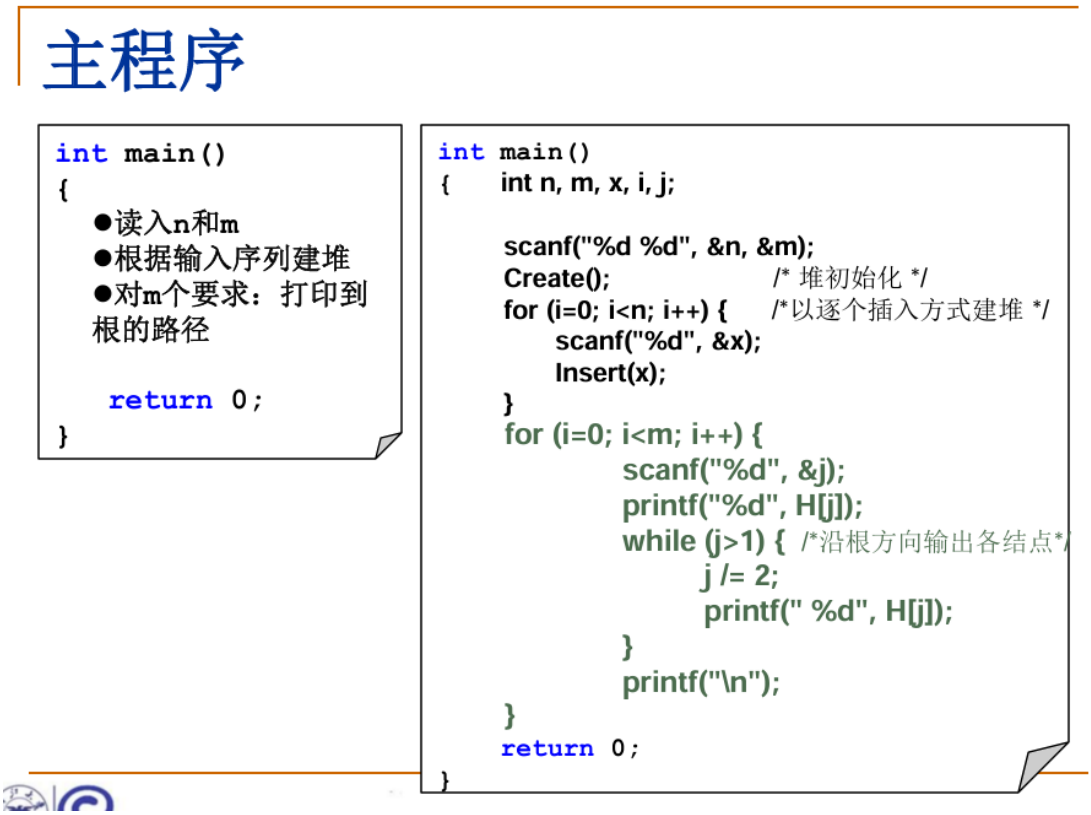
```
def insert_heap(heap, value):
    """向最小堆中插入一个新值

    Args:
        heap: 当前堆
        value: 要插入的值
    """
    heap.append(value)
    sift_up(heap, len(heap) - 1)


def sift_up(heap, i):
    """上浮操作，维护最小堆性质

    Args:
        heap: 当前堆
        i: 要上浮的节点索引
    """
    while i > 0:
```

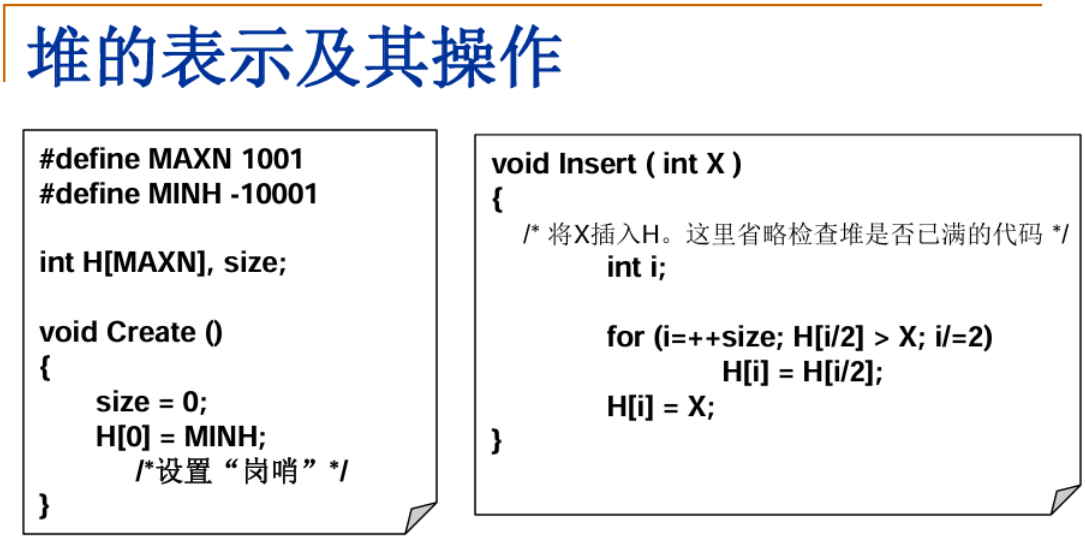
```
        parent = (i - 1) // 2
        if heap[i] >= heap[parent]:
            break
        heap[i], heap[parent] = heap[parent], heap[i]
        i = parent


def build_min_heap(nums):
    """从数组构建最小堆

    Args:
        nums: 输入数组

    Returns:
        构建好的最小堆
    """
    heap = []
    for num in nums:
        insert_heap(heap, num)
    return heap


def get_path_to_root(heap, index):
    """获取从指定节点到根节点的路径

    Args:
        heap: 当前堆
        index: 节点索引(0-indexed)

    Returns:
        从当前节点到根节点的路径(字符串形式)
    """
    path = []
    node = index
    while node >= 0:
        path.append(str(heap[node]))
        if node == 0:  # 到达根节点
            break
        node = (node - 1) // 2
    return " ".join(path)


def process_queries(heap, queries):
    """处理所有查询

    Args:
        heap: 当前堆
        queries: 查询索引列表(1-indexed)
    """
    for k in queries:
        # 将1-indexed转换为0-indexed
        index = k - 1

        # 检查索引是否有效
        if index < 0 or index >= len(heap):
            print(f"索引 {k} 超出范围")
            continue

        # 输出从当前节点到根节点的路径
        print(get_path_to_root(heap, index))


def main():
    """主函数，处理输入和输出"""
    # 读取输入
    n, m = map(int, input().split())
    nums = list(map(int, input().split()))[:n]
    queries = list(map(int, input().split()))[:m]

    # 建立最小堆
```

```
    heap = build_min_heap(nums)

    # 处理查询
    process_queries(heap, queries)


if __name__ == "__main__":
    main()
```

【整体思路】

代码结构与课程中基本一致





【关键点解析1】

```
    for k in queries:
```

```
        # 将1-indexed转换为0-indexed
        index = k - 1

        # 检查索引是否有效
        if index < 0 or index >= len(heap):
            print(f"索引 {k} 超出范围")
            continue

        # 输出从当前节点到根节点的路径
        print(get_path_to_root(heap, index))
```

需要注意的是，与C语言的实现方案不同，在本方案中是没有“哨兵”的，所以输入数据存在错位（即为实际列表中索引+1）

因此需要增加一步特殊处理