

# 1. E1 C 小水獭和多路归并

## 1.1. 题目描述

小水獭正在学习归并排序，它觉得将两个序列进行归并太简单了，因此想要尝试将多个序列进行归并，即多路归并。

具体来说，给定  $kk$  个长度为  $n$  的序列

$[a_{1,1}, a_{1,2}, \dots, a_{1,n}], [a_{2,1}, a_{2,2}, \dots, a_{2,n}], \dots, [a_{k,1}, a_{k,2}, \dots, a_{k,n}]$ ，满足每个序列均是严格递增的，且这  $kn$  个数互不相同。小水獭会重复进行如下操作直到所有序列均为空：

- 对于所有非空序列的首个元素，找到最小的那个元素，并将其从所在序列中删除。

小水獭会进行  $kn$  次操作，它想请你帮它写一个程序模拟这个过程，输出每一次操作删掉了哪个数。

## 1.2. 题解思路

**暴力做法：**

循环  $kn$  次，每次循环选出  $k$  个序列中最小的数，时间复杂度是  $k * n * k$ ，即  $10^{10}$  左右的时间复杂度，而本题时间复杂度要求是 1000ms，而这样一般会超时的。可参考一个时间复杂度的表（时间限制 1~2s）：

$n \leq 30$ , 指数级别, dfs+剪枝, 状态压缩dp

$n \leq 100 \Rightarrow O(n^3)$ , floyd, dp

$n \leq 1000 \Rightarrow O(n^2), O(n^2 \log(n))$ , dp, 二分

$n \leq 10000 \Rightarrow O(n\sqrt{n})$ , 块状链表

$n \leq 100000 \Rightarrow O(n \log(n)) \Rightarrow$  各种sort, 线段树、树状数组、set/map、heap、dijkstra+heap、spfa、求凸包、求半平面交、二分

$n \leq 1000000 \Rightarrow O(n)$ , 以及常数较小的  $O(n \log n)$  算法  $\Rightarrow$  hash、双指针扫描、kmp、AC自动机, 常数比较小的  $O(n \log n)$  的做法: sort、树状数组、heap、dijkstra、spfa

$n \leq 10^7 \Rightarrow O(n)$ , 双指针扫描、kmp、AC自动机、线性筛素数

$n \leq 10^9 \Rightarrow O(\sqrt{n})$ , 判断质数

$n \leq 10^{18} \Rightarrow O(\log n)$ , 最大公约数

从暴力做法中可见，其实我们需要的是每次取一个最小值，不过需要一个  $nk \log n$  的做法，所以就想到了使用堆来做。

**堆做法：**

我们需要维护的是由这  $k$  个序列的第一个元素组成的小根堆。

只要堆不为空，我们每次弹出堆顶的数，这一定是剩下所有数的最小值，接着把这个序列的下一个数压进堆中。重复这个操作，直至堆为空，输出的序列就是最小值了。

比如 -1, 3; -2, 4 这组数据，在读数时就先将 -1, -2 压入堆中，接着先弹出 -2，再把 4 压入堆中，弹出 -1，把 3 压入堆中，弹出 3，到达序列尾不再压，弹出 4。

**排序法：**

堆做法其实是对题意的一个简单模拟。

这道题其实仔细读题就能发现，最后输出的序列就是将所有数从小到大的排列。所以可以直接将所有数读进来，然后用 sort 或者 qsort 或其他  $n \log n$  排序方法即可。

## 1.3. 代码

```
#include <iostream>
#include <cstring>
#include <algorithm>
#include <cmath>
#include <queue>

using namespace std;
typedef long long ll;
typedef unsigned long long ull;
typedef pair<int, int> PII;

const int N = 1e5 + 10;
const int INF = 1e9;
int a[N];
int p[N];

int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);

    int T;
    cin >> T;
    while (T -- )
    {
        int cnt = 1;
        int k, n;
        cin >> k >> n;
        for (int i = 1; i <= n * k; i ++ )
        {
            cin >> a[i];
            if (n == 1 || i % n == 1) h.push({a[i], i});
        }

        while (!h.empty())
        {
            PII tmp = h.top();
            h.pop();

            cout << tmp.first << " ";

            int pos = tmp.second;
            if (pos % n != 0)
            {
                h.push({a[pos + 1], pos + 1});
            }
        }
        cout << "\n";
    }
    return 0;
}
```