

NOIP 普及组复赛 C 类题解思路(C++)

-----2011 T3

瑞士轮 `swiss.cpp`

在双人对决的竞技性比赛，如乒乓球、羽毛球、国际象棋中，最常见的赛制是淘汰赛和循环赛。前者的特点是比赛场数少，每场都紧张刺激，但偶然性较高。后者的特点是较为公平，偶然性较低，但比赛过程往往十分冗长。

本题中介绍的瑞士轮赛制，因最早使用于 1895 年在瑞士举办的国际象棋比赛而得名。它可以看作是淘汰赛与循环赛的折衷，既保证了比赛的稳定性，又能使赛程不至于过长。

$2*N$ 名编号为 $1 \sim 2N$ 的选手共进行 R 轮比赛。每轮比赛开始前，以及所有比赛结束后，都会按照总分从高到低对选手进行一次排名。选手的总分为第一轮开始前的初始分数加上已参加过的所有比赛的得分和。总分相同的，约定编号较小的选手排名靠前。

每轮比赛的对阵安排与该轮比赛开始前的排名有关：第 1 名和第 2 名、第 3 名和第 4 名、……、第 $2K - 1$ 名和第 $2K$ 名、……、第 $2N - 1$ 名和第 $2N$ 名，各进行一场比赛。每场比赛胜者得 1 分，负者得 0 分。也就是说除了首轮以外，其它轮比赛的安排均不能事先确定，而是要取决于选手在之前比赛中的表现。

现给定每个选手的初始分数及其实力值，试计算在 R 轮比赛过后，排名第 Q 的选手编号是多少。我们假设选手的实力值两两不同，且每场比赛中实力值较高的总能获胜。

输入

第一行是三个正整数 N 、 R 、 Q ，每两个数之间用一个空格隔开，表示有 $2*N$ 名选手、 R 轮比赛，以及我们关心的名次 Q 。

第二行是 $2*N$ 个非负整数 s_1, s_2, \dots, s_{2N} ，每两个数之间用一个空格隔开，其中 s_i 表示编号为 i 的选手的初始分数。

第三行是 $2*N$ 个正整数 w_1, w_2, \dots, w_{2N} ，每两个数之间用一个空格隔开，其中 w_i 表示编号为 i 的选手的实力值。

输出

只有一行，包含一个整数，即 R 轮比赛结束后，排名第 Q 的选手的编号。

样例输入 1

```
2 4 2
7 6 6 7
10 5 20 15
```

样例输出 1

```
1
```

数据范围

对于 30% 的数据， $1 \leq N \leq 100$ ；

对于 50% 的数据， $1 \leq N \leq 10,000$ ；

对于 100% 的数据， $1 \leq N \leq 100,000$ ， $1 \leq R \leq 50$ ， $1 \leq Q \leq 2N$ ，

$0 \leq s_1, s_2, \dots, s_{2N} \leq 108$ ， $1 \leq w_1,$

$w_2, \dots, w_{2N} \leq 108$ 。

解 析

1、这道题似乎很简单，每一轮之前都对选手排序，之后根据选手实力算出得分，经过 R 轮后自然求出答案。对于第三题，似乎有点简单的有点过分。

2、看一下数据范围，测算一下时间复杂度： $O(R*(N*\log N+N))$ ，注意快排时间复杂度 $N*\log N$ ，因为不稳定，最糟糕情况是 $R*N^2$ 。

3、本题的正解是运用分治算法的归并排序，在输入后先用 `sort` 排序一次，然后，每进行一轮比赛，进行一次归并排序，归并排序时，将赢的选手得分加 1，并放入赢的数组（结构体），将输的选手放入输的数组（结构体），然后就直接将赢和输两数组归并进原数组中，最后输出答案即可。这种算法的时间复杂度为 $O(R*2*N)$ ，不会超时。需要注意的是，在归并排序时，也需要注意选手得分相同时，将编号小的排在前面。