NOIP模拟赛2024

Maybe a Simulation

题目名称	战队分配	货车运输	甜果	打平就能出线！
目录	team	cargo	sugar	qualify
可执行文件名	team	cargo	sugar	qualify
输入文件名	team.in	cargo.in	sugar.in	qualify.in
输出文件名	team.out	cargo.out	sugar.out	qualify.out
每个测试点时限	2.0 秒	1.0 秒	1.0 秒	4.0 秒
内存限制	1 GB	1 GB	1 GB	1 GB
子任务数目	2	3	6	7
是否捆绑测试	是	是	是	是

提交源程序文件名

对于 C++ 语言	rally.cpp	tree.cpp	sugar.cpp	qualify.cpp

编译选项

对于 C++	-lm -std=c++14 -O2 -Wl,--stack=998244353

【注意事项（请仔细阅读）】

1. 选手提交的源程序请**直接放在个人目录下**，无需建立子文件夹；
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
4. **对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理**。
5. 若无特殊说明，结果比较方式为**忽略行末空格、文末回车后的全文比较**。。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 在终端中执行命令 `ulimit -s unlimited` 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
8. 若无特殊说明，每道题的**代码大小限制为 100KB**。
9. 若无特殊说明，输入与输出中同一行的相邻整数、字符串等均使用一个空格分隔。
10. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 scanf 函数）避免出错。

11. 直接复制 PDF 题面中的跨页样例，数据将带有页眉页脚，建议选手直接使用对应目录下的样例文件进行测试。
12. 使用 std::deque 等 STL 容器时，请注意其内存空间消耗。
13. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外**不允许在程序中手动开启其他编译选项**，一经发现，本题成绩以 0 分处理。

战队分配（team）

【题目描述】

你和你的好友小 W 正在为一场比赛准备战队分配方案。这场比赛包含多个玩家，你需要将玩家们分成两组。刚看过国足 0:7 日本的你看热闹不嫌事大，决定让两组之间的实力差距尽可能大。

对于每两个玩家 i 和 j，有一个默契度 $w_{i,j}$，表示当他们在同一组中时对该组贡献的战力。每组的实力是队内所有无序玩家对 (i,j) 的 $w_{i,j}$ 之和。

分成的两组大小相等 ,保证n是偶数。

【输入格式】

从文件 ***team.in*** 中读入数据。第一行包含一个偶数 n，表示玩家的总数。接下来的 n 行，第 i 行包含 n 个整数 $w_{i,1}, w_{i,2}, \ldots, w_{i,n}$，表示玩家之间的默契度。保证 $w_{i,i} = 0$ 且 $w_{i,j} = w_{j,i}$。

【输出格式】

输出到文件 ***team.out*** 中。输出两个组之间的最大可能实力差距。

【样例输入 1】

```
6
0 4 -6 2 3 -3
4 0 2 -6 0 0
-6 2 0 0 2 2
2 -6 0 0 -1 5
3 0 2 -1 0 -4
-3 0 2 5 -4 0
```

【样例输出 1】

```
0
```

【样例输入 2】

```
4
0 1 2 2
1 0 8 -3
2 8 0 5
2 -3 5 0
```

【样例输出 2】

```
6
```

【测试点约束】

对于所有测试点，满足以下约束：

$2 \leq n \leq 1000$，$-10^6 \leq w_{i,j} \leq 10^6$。

保证 $w_{i,i} = 0$ 且 $w_{i,j} = w_{j,i}$。

每个子任务的具体限制见下表：

子任务编号	分值	$n \leq$
1	20	20
2	80	1000

货车运输（cargo）

【题目描述】

某地有两家运输公司，这两家公司各自拥有 n 辆货车，运载能力分别为 1 到 n。

但是两家公司的车队排班都由总部统一安排，每天，每家公司只会向此地派一辆车。小 W 并不知道每天具体的运力是多少，只知道在未来 n 天内，每家公司的 n 辆车各会前来恰好一次。

小 W 每天都会有货物需要运输，货物不可拆分，必须整体装入某一辆货车的车厢。因此，第 i 天的运输能力为当天两家公司派出的两辆车中运载能力的较大值，即 $max(p_i, q_i)$，其中 p 和 q 是两家公司货车排班的排列。

小 W 现在想要知道，在所有可能的排班方案中，有多少种排班方案可以使得未来 n 天的总运输能力恰好等于 k。答案对 $10^9 + 7$ 取模。

【输入格式】

从文件 ***cargo.in*** 中读入数据。第一行包含两个整数 n 和 k。

【输出格式】

输出到文件 ***cargo.out*** 中。输出符合条件的排班方案数量。答案对 $10^9 + 7$ 取模。

【样例输入 1】

```
1 2 4
```

【样例输出 1】

```
1 2
```

【样例输入 2】

```
1 3 7
```

【样例输出 2】

```
1 12
```

【测试点约束】

对于所有测试点，满足以下约束：

$$1 \le n \le 100$$

$$1 \le k \le n^2$$

每个子任务的具体限制见下表：

子任务编号	分值	$n \le$	特殊限制
1	20	10	
2	5	100	$k = \frac{n(n+1)}{2}$
3	75	100	

甜果（sugar）

【题目描述】

在一个偏远的山村里，村民们依靠种植甜果树为生。每年丰收时，村长会组织一场分发甜果的活动，作为村里的传统庆典。

今年，村子里有 n 个孩子参加了这场活动。每个孩子最初收获了 a_i 包甜果。在庆典上，村长会以随机顺序揭晓 n 次特殊的“赠果”事件，规则如下：

第 i 次赠果事件时，如果第 i 个孩子的甜果包数**严格多于**第 b_i 个孩子的甜果包数，那么第 i 个孩子将额外获得 w_i 包甜果。否则，什么也不会发生。

由于赠果事件的顺序是随机的，孩子们都想知道，当所有事件完成后，自己预计能拥有多少包甜果。

可以证明答案可以表示为一个不可约分的分数 $\frac{x}{y}$，其中 x 和 y 是整数，并且 $y \not\equiv 0 \pmod{10^9+7}$。输出等于 $x \cdot y^{-1} \pmod{10^9+7}$ 的整数。换句话说，输出这样一个整数 a，使得 $0 \le a < 10^9+7$ 并且 $a \cdot y \equiv x \pmod{10^9+7}$。

【输入格式】

从文件 ***sugar.in*** 中读入数据。

每个测试包含多组数据。第一行包含一个单一整数 t $(1 \le t \le 5 \cdot 10^5)$ 表示数据组的数量。对于每组数据：

第一行包含一个单一整数 n $(1 \le n \le 5 \cdot 10^5)$ 表示孩子的数量。

第二行包含 n 个整数 a_i $(1 \le a_i \le 10^9)$：每个孩子最初拥有的甜果包数。

第三行包含 n 个整数 b_i $(1 \le b_i \le n)$。

第四行包含 n 个整数 w_i $(1 \le w_i \le 10^9)$。

保证所有数据组中 n 的总和不超过 $5 \cdot 10^5$。

【输出格式】

输出到文件 ***sugar.out*** 中。

对于每组数据，输出一行 n 个整数：每个孩子将拥有的甜果包数的期望值。如上所述，以模 10^9+7 的方式输出答案。

【样例输入】

```
4
4
2 5 5 2
4 2 1 3
3 2 1 4
3
5 4 3
1 1 1
6 6 6
3
5 4 3
2 3 1
1 2 3
5
2 1 3 2 1
5 1 1 3 4
1 3 4 2 4
```

【样例输出】

```
2 5 6 2
5 4 3
500000009 6 3
3 1 5 2 1
```

【测试点约束】

对于所有测试点的约束，见输入格式。

每个子任务的具体限制见下表：

子任务编号	分值	$\sum n \leq$	特殊限制
1	5	$5 \cdot 10^5$	$b_i = i$
2	10	10	
3	10	100	
4	25	1000	
5	10	$5 \cdot 10^5$	$b_i = i \mod n + 1$
6	40	$5 \cdot 10^5$	

打平就能出线！（qualify）

【题目描述】

「xx 队本场比赛打平即可提前两轮出线！」

「xx 队若最后三轮全胜，仍然有理论出线希望！」

「xx 队基本确定提前两轮保级！」

足球媒体热衷算分。然而，算分并非易事，这些分析可能出错。赛事中存在多队连环套，各队已赛场次也可能不同，情况远比直觉复杂。

考虑这样一个极端案例：

队伍	场次	积分	净胜球
A 队	4	6	0
B 队	2	6	+5
C 队	2	3	0
D 队	2	3	0
E 队	2	0	-5

假设 A 队已经完成所有比赛，剩余的比赛安排为：B 队对阵 C 队，C 队对阵 D 队，D 队对阵 E 队，E 队对阵 B 队。小组前四名将晋级。

每场比赛胜者得 3 分，负者得 0 分，平局各得 1 分。

此时，A 队似乎接近提前出线，但其实无法确保获得出线资格。如果剩余比赛结果如下：

- B 队 0:1 C 队
- C 队 0:1 D 队
- D 队 0:1 E 队
- E 队 0:1 F 队
- B 队 0:4 F 队

那么，所有五支队伍将同为 6 分，0 净胜球，A 队将可能被淘汰。

因此，小 W 想知道，给定赛季末的积分榜，一支特定队伍的最高和最低可能排名究竟是多少。

为简化起见，我们只考虑足球排名规则中的积分与净胜球两项，队伍积分相同时，即比较净胜球；净胜球再相同时，即抽签决定名次。

由于现在已经赛季末，保证每支队伍剩余比赛不超过 2 场。

【输入格式】

从文件 ***qualify.in*** 中读入数据。

第一行包含三个正整数 n, m, d ，表示共有 n 队参赛，每队均会进行 m 场比赛，小 W 关注的是 d 号队伍。

接下来的 n 行，每行包含两个非负整数 c_i, p_i ，一个整数 gd_i，表示 i 号队已经完成了 c_i 场比赛，获得 p_i 分和 gd_i 个净胜球。

接下来若干行，每行包含两个正整数 x_i, y_i ，表示剩余比赛中有一场 x_i 号队与 y_i 号队的比赛。

保证剩余的比赛进行后，每支队伍均进行了 m 场比赛。

【输出格式】

输出到文件 ***qualify.out*** 中。

输出两行两个正整数 U, D ，表示 d 号队伍最高排名第 U 名，最低排名第 D 名。

【样例输入】

```
5 4 1
4 6 0
2 6 5
2 3 0
2 3 0
2 0 -5
2 3
3 4
4 5
2 5
```

【样例输出】

```
1
5
```

【样例解释】

样例数据给出的是题目描述中的积分榜。在五支队伍均为 6 分，0 净胜球时，A 队可能在抽签中获得任何名次。

【测试点约束】

对于所有测试点：$1 \le n, m \le 3 \cdot 10^5$，$1 \le d \le n$ ，$m-2 \le c_i \le m$，$0 \le p_i \le 3m$，$-10^8 \le gd_i \le 10^8$，$x_i \ne y_i$。

保证没有两队在剩余比赛中交手两次。

不保证题目中给出的积分榜可以由实际比赛组合达到。

每个测试点的具体限制见下表：

子任务编号	分值	$n \le$	$m \le$	特殊限制
1	10	6	6	
2	5	1000	1000	$m - c_i \le 1$
3	10	1000	1000	
4	15	$3 \cdot 10^5$	$3 \cdot 10^5$	$m - c_i \le 1$
5	60	$3 \cdot 10^5$	$3 \cdot 10^5$	