T1 gio

50 pts

对于 50% 的数据,有 $n \le 2$ 或 $m \le 2$。

暴力枚举或手动判断即可。

100 pts

对于 100% 的数据，$1 \le n \le 1000$，$1 \le m \le 10000$。

可以推理出一个结论，总分 A 一定比总分 B 高且仅当 A 的每一题都比 B 高。

于是我们只要统计最多有多少人至少一场比赛比 $Luke$ 高，最多有多少人至少一场比赛比 $Luke$ 低即可。

答案为 $m - \min(m-1, \sum m - r_i)$ 和 $1 + \min(m-1, \sum r_i - 1)$。

T2 girl

50 pts

我们很容易得到一个贪心的算法：把最小的 A 和最大的 B 配对，第二小的 A 和第二大的 B 配对 ...... 最大的 A 和最小的 B 配对。

这事实上是一个插入排序，每次 $O(n)$ 判断，最终复杂度 $O(n^2)$

100 pts

注意到所有 A 和 B 不超过100。

利用这个条件，我们维护列表 $totA$：$totA[i]$ 表示大小为 i 的 A 有多少个

同理定义 $totB$

这样在得到新的 A, B 时，我们可以 $O(1)$ 的将其插入。

在回答询问时，我们只需维护两个指针 p 和 q，设 $totA[p]$ 和 $totB[q]$ 的较小者为 T，将两者都减去 T，得到 T 对 $(totA[p]，totB[q])$，然后继续移动两个指针即可。

时间复杂度 $O(100n)$。

T3 stone

20 pts

对于20%的数据，有 $1 \le T \le 5, 1 \le n, m \le 5$。

暴力搜索判断即可。

50 pts

对于50%的数据，有 $1 \le T \le 5, 1 \le n \le 10^6$。

递推：

在不考虑环形即不考虑 A_1 与 A_n 是否同色时的方案数有 $m(m-1)^{n-1}$。将上面方案数减去 A_1 与 A_n 同色的方案数即把这两块当成同一区域，此时方案数为 a_{n-1}。

$$a_n = (m-1+1)(m-1)^{n-1} - a_{n-1} \quad (n \geq 2)$$

$a_n = (m-1)^n + (m-1)^{n-1} - a_{n-1} \quad (n \geq 2)$

$a_n - (m-1)^n = -[a_{n-1} - (m-1)^{n-1}]$

$b_{n-1} = -b_n$

$b_n = a_n - (m-1)^n$ 这个数列是一个公比为 -1 的等比数列

$b_2, b_3, b_4, b_5, \ldots., b_n$ 公比为 -1 的等比数列

$b_2 = a_2 - (m-1)^2 = m*(m-1) - (m-1)^2 = m-1$

$a_n = (m-1) \times (-1)^{n-2} + (m-1)^n$

$n = 2$ 代入 $a_2 - (m-1)^2 = m*(m-1) - (m-1)^2 = m-1$

首项为 $m-1$ 公比为 -1 的等比数列

$a_n = (-1)^{n-2}(m-1) + (m-1)^n$

100 pts

对于100%的数据，有 $1 \leq T \leq 10^5, 1 \leq n, m \leq 10^9$。

由上可知 $a_n = m(m-1)^{n-1} - a_{n-1}$

整理得 $a_n - (m-1)^n = -[a_{n-1} - (m-1)^{n-1}]$

令 $b_n = a_n - (m-1)^n$ 则 $\{b_n\}$ 是公比为 -1 首项 $b_2 = m(m-1) - (m-1)^2 = m-1$ 的等比数列：

$$b_n = (-1)^{n-2}(m-1) \quad (n \geq 2)$$

所以 $a_n = (m-1)^n + (-1)^{n-2}(m-1) \quad (n \geq 2)$

游览计划

- 考点：**图论**，**bfs**，贪心。
- 难度：CSP-J T4

测试点 1~3

给暴力留的分。

测试点4~8

先通过bfs求出每个点到另一个点的最短路径的长度 $d_{i,j}$ ，然后暴力枚举四个点 a, b, c, d ，然后求 $\max(d_{a,b} + d_{b,c} + d_{c,d})$ 就可以了。

复杂度 $O(n^4)$ 。

测试点9~10

答案就是 3，因为在完全图中，任意两点的最短路都是 1 。

测试点11~15

同样求出 $d_{i,j}$ ，然后暴力枚举三个点 a,b,c ，然后求出离 c 点最远的点 f_c，但是 f_c 可能是 a 或者 b ，所以需要求出离 c 点第二远的点 g_c，第三远的点 h_c ，这样就一定可以找出那个最远的点。

复杂度是 $O(n^3)$ 。

当然也有许多其他做法。

测试点16~20

优化上一个算法。

同样求出 $d_{i,j}$ 和 f_i, g_i, h_i 。

我们枚举两个点 a,b ，然后经过 $a \to b$ 的最远路径一定是 $f_a/g_a/h_a \to a \to b \to f_b/g_b/h_b$ 其中之一。

依次考虑这几种情况，排除掉非法的情况（经过相同的点），直接枚举就可以了。

复杂度是 $O(n^2)$ 。