

智商锁(lock)

Time Limit:1000ms Memory Limit:128MB

题目描述

LYK 想把你的智商锁起来，它出了道神题。

它有一壶水，这壶水的水量在 $[L, R]$ 中，你只知道 L 和 R 是多少，但你并不知道具体的值是多少。

它有两个容量为无穷大的杯子。它想尽可能把壶里的水均匀的分在两个杯子中。它觉得这个任务可能太难了，因此它只要求最终两个杯子的水的容量至多相差 1。并且它也不要求把壶里的水全部倒完，只要求最终剩在壶里的水不超过 1。

LYK 在倒水过程中，不知道这壶水还有多少水，但能知道是否倒完了。

一个最笨的方法是左边倒 1 滴，右边的倒 1 滴，左边再倒 1 滴，这样依次下去，直到壶里没水或者已经倒了 $R-1$ 的水时，就满足条件了。

但根据题名，这题并不简单。

LYK 想知道如果采用最优策略，则在最坏情况下 LYK 要倒多少次水才能满足条件。

(注：最优策略下的最坏情况指，事件总是往最坏的情况发展，例如 $L=2, R=10000$ ，你第一次就大胆的往第一杯水倒 2 滴水，但可能壶里一开始总共就只有 2 的水，就不满足条件了。所以你的方案应当是就算情况再坏，也能满足条件，并且在此基础上倒水的次数最少)

输入格式(lock.in)

一行两个数 L, R 。

输出格式(lock.out)

一个数表示答案。

输入样例 1

2 5

输出样例 1

2

输入样例 2

2 2

输出样例 2

1

样例解释：

样例 1：先在左边的杯子倒 1.5 滴水（没错不一定是整数！），把剩下的水都倒进右边的杯子里直到右边的杯子水量为 2.5 或者壶里没水就可以了。

样例 2：只要在左边的杯子倒 1 滴，就可以了。

数据范围

对于 30%的数据， $R \leq 5$

对于另外 20%的数据 $R-L \leq 5$

对于 70%的数据 $R \leq 1000$

对于 100%的数据 $1 \leq L \leq R \leq 10^{16}$ 。

LYK 与序列 (sequence)

Time Limit:1000ms Memory Limit:128MB

题目描述

LYK 有一个序列，它想玩一个游戏，就是找一段最长的连续的数字，使得它们的和小于等于 p ，我们假设这个最长的区间的长度为 L 。

但这件事好像有点简单，因此 LYK 给自己一次机会：将其中一段连续的长度不超过 d 的区间的数都变成 0，之后再找最长的连续的数字使得和小于等于 p 。

LYK 想知道，在所有的可能的改变策略中， L 最大为多少。

输入格式(sequence.in)

第一行三个数 n, p, d ，表示有一个长为 n 的序列， p 和 d 的定义在和题目描述中一样。

接下来一行 n 个数 a_i ，表示一开始的序列。

输出格式(sequence.out)

一个数表示答案。

输入样例

```
9 7 2
3 4 1 9 4 1 7 1 3
```

输出样例

```
5
```

样例解释

将 $[4, 5]$ 这段区间对应的数变成 0，之后可以选 $[2, 6]$ ，长度为 5。

数据范围

对于 20% 的数据 $n \leq 100$ 。

对于 40% 的数据 $n \leq 1000$ 。

对于另外 20% 的数据 $d=1$ 。

对于 80% 的数据（包括 $d=1$ 的 20%）， $n \leq 100000$ 。

对于 100% 的数据 $1 \leq d \leq n \leq 10^6$ ， $0 \leq p \leq 10^{16}$ ， $0 \leq a_i \leq 10^9$ 。

建议使用早上下发的读入优化

打架 (fight)

Time Limit:1000ms Memory Limit:128MB

题目描述

LYK 有 n 个小朋友排成一排。第 i 个小朋友的战斗力的 a_i ，且他们的战斗力互不相同。战斗力高的会打败战斗力低的。

LYK 想恶搞这些小朋友们，具体地，它有 k 次操作。

第 i 次操作会有两个参数 l_i 和 r_i ，表示如果两个小朋友 A, B 的战斗力的均在 $[l_i, r_i]$ 这段区间中，它们的打架结果会相反。即如果一开始 A 能赢 B，则现在变成 B 能赢 A。当然它们的打架结果可能在后来的操作中又被反过来。

LYK 想知道， k 次操作后，存在多少三元组 (a, b, c) ，其中 a 能赢 b ， b 能赢 c ， c 能赢 a 。注意这里 (a, b, c) ， (b, c, a) ， (c, a, b) 算同一种三元组。

输入格式 (fight.in)

第一行两个数 n, k 。

第二行 n 个数表示 a_i 。

接下来 m 行，每行两个数 l_i, r_i 。

输出格式 (fight.out)

一个数表示答案。

输入样例

3 2

1 2 3

1 2

2 3

输出样例

1

样例解释

进行过操作后，1 能赢 2，2 能赢 3，而 3 一开始就能赢 1 并且结果没被改变过，所以就存在 1 个符合条件的三元组。

数据范围

对于 20% 的数据 $n, k \leq 100$ 。

对于 60% 的数据 $n, k \leq 1000$ 。

对于另外 10% 的数据 $k=0$ 。

对于 100% 的数据， $3 \leq n \leq 10^5$ ， $0 \leq k \leq 10^5$ ， $0 \leq a_i, l_i, r_i \leq 10^9$ ， $l_i \leq r_i$ 。