

THUWC/PKUWC 2018 膜你赛

——lych_cys

试题名称	乘船旅行	掘地求升	句读之王
程序名	armstrongboat	gettingoverit	mandarin
测试点数目	20	20	20
单个测试点分值	5	5	5
时间限制	1s	3s	5s
空间限制	2048M	2048M	2048M
题目类型	传统题	传统题	传统题

评测打开 O2 优化，无栈空间限制

注意读入的规模和常数因子可能带来的影响

乘船旅行 (armstrongboat.pas/c/cpp)

1s/2048M

【问题描述】

有 $2N$ 个人乘船游玩，他们的重量用一个数组 w 表示。但是他们还没有选择应该坐什么船。现在他们看中了一条阿姆斯特朗回旋加速喷气式阿姆斯特朗船，船的长度为 n 。由于船左右两侧都能坐人，因此船上的位置既不会多，也不会少。

由于每个人喜好不同，他们喜欢坐的位置也有所不同。每一个人会有恰好两个他愿意坐的位置（可能在同侧，也可能在异侧）。

如果船两侧的重量之差超过了 K ，那么就会有翻船的危险。

现在你想知道，如果租用这条阿姆斯特朗回旋加速喷气式阿姆斯特朗船，能否在保证安全的前提下，使所有人都能坐在他们喜欢的位置上？如果能，请求出船两侧的重量之差的最小值，或者给出方案数对 998244353 取模的值。（两个方案不同当且仅当存在一个人，他坐的位置不同）。

【输入格式】

第一行两个正整数 N, K, op ，表示船的长度 N ，船两侧重量之差的最大值 K ， op 表示询问的类型。

接下来 $2N$ 行，每行 5 个非负整数 $pi1, xi1, pi2, xi2, wi$ ，其中：

(p_{i1}, x_{i1}) 和 (p_{i2}, x_{i2}) 表示第 i 个人喜欢坐的两个位置，保证 $p_{i1} \neq p_{i2}$ 或 $x_{i1} \neq x_{i2}$ 。 p_i 为 0 表示左，为 1 表示右。 x_i 表示这个位置位于从船头起的第 x_i 个。

w_i 表示第 i 个人的重量。

【输出格式】

第一行一个字符串 YES 或 NO，表示是否符合题目要求。

如果第一行为 YES，输出的第二行应给出一个数 x ，表示最小差值

($op=0$)或方案数对 998244353 取模的值($op=1$)

【样例输入 1】

4 1 1

0 1 1 1 1

0 2 1 1 2

0 2 1 2 8

0 1 1 2 2

0 3 1 3 5

0 3 1 3 2

0 4 1 4 1

0 4 1 4 2

【样例输出 1】

YES

2

【样例输入 2】

2 5 0

0 1 1 1 1

0 1 1 2 4

0 2 1 2 1

0 2 1 1 4

【样例输出 2】

NO

【样例输入 3】

2 2 0

0 1 0 2 2

0 1 0 2 1

1 1 1 2 2

1 1 1 2 2

【样例输出 3】

YES

1

【数据规模】

T	$N \leq$	$\sum w_i \leq$	op=	特殊性质
1	10	500000	1	
$2 \sim 6$	1000	5000		
$7 \sim 8$	50000	500000		保 证 $w_1=w_2=\dots w_{2n}$
$9 \sim 13$		200000	0	
$14 \sim 20$		500000	1	

对于 100%的数据，保证 $1 \leq N \leq 50000$ ， $1 \leq K \leq \sum w_i \leq 500000$ ， $op \in \{0, 1\}$

掘地求升(gettingoverit.pas/c/cpp)

3s/2048M

【问题描述】

你在 ois 手机应用商店 pap sotre 下载了一款名为“掘地求升”的游戏。

游戏的地图可以抽象为下底面周长为 N ($N>1$)，高度为 M ($M>1$) 的圆柱体。你要在圆柱的外表面上努力攀登。圆柱体的外表面被分成 $N*M$ 个方格，每个方格上有一些金币，按照自下而上，从上看逆时针的方向给出每个位置上金币的个数 ($A_{i,j}$)。你可以任意选择第一行的某个格子作为起点，而每一步可以选择向上、左上或右上攀登一步，到达第 M 层时游戏结束。假设你每一步获得金币数量的积为 X 。

同时，这个圆柱体还是一个好看的圆柱体。圆柱体的每一列被涂成了红(r)，蓝(b)，白(w)三种颜色之一。在你进行游戏时有两个奖励参数 a, b ，初始值为 1。每当你经过一个红色的格子（即这个格子所在的列被涂成了红色）， a 的值增加 1；每当你经过一个蓝色的格子， b 的值增加 1。

为了鼓励玩家打出高(水)ping 的操作，游戏的奖励值还与 Combo 有关。Combo 指的是所有极长相同连续操作长度的乘积，具体地说：

- 用 U, L, R 分别表示向上、左上、右上攀登一步的操作，按照操作顺序我们可以得到一个字符串（如果我们的操作是：上、上、左上、右上、右上、右上、左上，那么对应的字符串就是 UULRRRL）

- 将得到的字符串求出所有极长子串的长度（如 UULRRRL 就是 (2, 1, 3, 1)）

- 那么 Combo 的值就是这些长度的乘积（UULRRRL 的 Combo 就是 $2*1*3*1=6$ ）

一局游戏结束后，你的得分就是 $X*a*b*Combo$ 。你并不知道怎么玩才能得到最高的分数，因此你决定每一步随机选择三个方向中的一个，那么最后你得到的分数的期望是多少呢？你只需要输出答案 $*N*3^M$ 对 998244353 取模的值。

【输入格式】

第一行两个正整数 N, M，表示圆柱体的周长和高。

第二行一个长度为 N 的字符串，表示每一列被涂成的颜色。保证仅由 r, b, w 组成。

接下来 M 行，每行 N 个正整数，第 i 行第 j 列表示从下往上第 i 行，从上方看逆时针的第 j 个格子中的金币个数。

【输出格式】

一个数，表示得分期望 $*N*3^M$ 对 998244353 取模的值。

【样例输入 1】

2 3

rb

1 1

1 1

1 1

【样例输出 1】

136

【样例输入 2】

3 3

rbw

1 2 3

4 5 6

7 8 9

【样例输出 2】

9324

【数据范围】

T	N≤	M≤	特殊限制
1	20	10	

2	50	50	
3~7	1000	1000	字符串只含有 w
8~10	300	300	保证 $a_{i,j}=1$
11	100	100	
12	300	300	
13~16	1000	1000	
17~18	1500	1500	
19~20	2000	2000	

对于 100%的数据， $N, M \leq 2000$ ， $1 \leq a_{i,j} \leq 1e9$

句读之王(mandarian.pas/c/cpp)

5s/2048M

【问题描述】

句读之王很喜欢句读，他的院子里有两棵树，一棵是橘子树，另一棵也是橘子树。句读之王比较喜欢第一棵橘子树，这棵橘子树有 N 个节点标号为 $1..N$ ，并通过 $N-1$ 根枝条连接成一个连通的图。一开始树上的每个节点都没有橘子。每经过 $1s$ ，第 i 个节点就会长出 a_i 颗新的橘子。每个节点的橘子数量也有一个上限 b_i ，如果某时刻某个节点的橘子个数超过上限，那么超过上限的橘子就会消失。

句读之王擅长把橘子变成豆子。如果某一个节点有 x 颗橘子，那么句读之王可以把它们变成 $f(x)$ 颗豆子。 $f(x)$ 定义为最大的完全平方数 y 满足 y 是 x 的因数。

可能在某些时刻，他会选择一些节点，对每一个节点，他会把每个节点上面的橘子全部变成豆子，然后摘下来，把得到的豆子喂给他的宠物海蜇吃。（注意：是先把每个节点的橘子变成豆子再摘下来，而不是先全部摘下来再变成豆子）

句读之王一共会进行 M 次操作，这些操作按照时间给出，且同一时刻他最多只能进行一次操作。现在他想知道，每一轮操作海蜇吃了几颗豆子。为了方便，你只需要输出所有答案的和即可。如果你不会做这道题，那么给句读之王买两个橘子，也是可以的。

【输入格式】

第一行两个正整数 N, M ，表示树上节点的个数和接下来操作的个数

接下来一行 N 个非负整数 a_i ，分别表示第 i 个节点每秒钟会生长出的橘子的个数

接下来一行 N 个非负整数 b_i ，分别表示第 i 个节点橘子数量的上限

接下来 $N-1$ 行每行两个正整数 x_i, y_i ，表示第 i 根枝条连接的两个节点

接下来 M 行每行三个正整数 X_i, Y_i, t_i ，表示选择所有位于 $X_i \sim Y_i$ 最短路上的节点，这个操作发生在第 t_i 秒。保证 $t_i < t_{i+1}$

【输出格式】

一个数，表示所有答案的和

【样例输入 1】

2 1

2 3

4 5

1 2

1 2 2

【样例输出 1】

5

【样例输入 2】

10 7

2 1 2 2 1 2 3 4 6 5

5 6 7 8 9 10 11 12 13 14

1 2

1 6

2 3

2 4

3 5

4 7

4 8

4 9

5 10

3 6 2

5 9 4

1 10 5

2 7 10

6 6 15

2 3 16

3 8 19

【样例输出 2】

51

【数据规模】

T	N≤	M≤	t_N≤	特殊性质
1	100	100	1000	
2	100000			
3~4				
5~11	100000	100000		300000
12~13	30000	30000		
14~15	100000	100000	满足 ai, bi≤100	
16~20				

对于 100%的数据，满足 $N, M \leq 100000, t_N \leq 300000, 0 \leq a_i \leq 100000,$
 $b_i \leq 2e10$