

骨牌(domino.cpp/domino.in/domino.out) 2s/512M

问题描述

有一个 $2N \times 2N$ 的网格，在格子中需要放一些 1×2 的骨牌。一个合法的方案满足任意 2×2 的单位中至少还能放入一张骨牌，并且骨牌的总数量尽可能大

现在有网格中已经放了一些骨牌，接下来有 M 次询问，每次询问时加入一张骨牌，你要求出之后有多少种放置骨牌的方法使得最终的结果合法，对 998244353 取模

每次询问结束后还原到初始状态

注意初始局面后可能出现无法达成合法状态的情况；“总数量尽可能大”指的是从空局面开始放牌的情况下的最大总数量

输入格式

第一行两个数 N, M

接下来 $2N$ 行每行一个长度为 $2N$ 的串表示初始局面，'D'表示骨牌

接下来 M 行每行 4 个数 $x1, y1, x2, y2$ ，表示骨牌的两个端点分别为 $(x1, y1), (x2, y2)$

输出格式

M 个数表示答案

样例输入

```
2 2
..DD
....
D...
D...
1 1 1 2
3 3 3 4
```

样例输出

4

2

数据范围

测试点编号	$N \leq$	$M \leq$
1	3	10
2	5	
3	10	
4	15	
5	100	5
6	100	100
7	1500	1000
8		300000
9		
10		

最后一个测试点满足初始局面没有骨牌

随机(random.cpp/random.in/random.out) 4s/1024M

问题描述

有一棵 N 个点的树和 M 个操作 x, y, S ，表示给 x 到 y 的链上的节点都加入一个数字字符串 S

所有操作都结束后需要对每个点进行一次询问。首先在该节点的所有字符串形成的 trie 上随机选择一个点（此时为步数 0），这里定义 trie 的根节点为空串，然后每次随机一个与当前点相邻的点移动过去且步数+1，如果移动到了某个字符串对应的节点则结束。询问每个点对应 trie 的期望步数。

输出模 998244353 意义下的值。假如答案为 $\frac{x}{y} (\gcd(x, y) = 1)$ ，那么需要输出

$r(0 \leq r < 998244353)$ 满足 $x \equiv y \times r \pmod{998244353}$

数据保证答案不会存在 $y \equiv 0 \pmod{998244353}$ 的情况

输入格式

第一行一个数 N

接下来 $N - 1$ 行，每行描述一条边

接下来一个数 M

接下来 $2M$ 行 x, y, S ，表示一次操作

输出格式

N 个数表示每个点对应的答案

样例输入

3

1 2

1 3

2

1 2

012

1 3

0

样例输出

499122177

499122182

499122177

数据范围

测试点范围	$N \leq$	$M \leq$	$\sum S \leq$	性质
1	10	30	30	
2	100	150	1000	
3				
4	1000	1000	3000	字符串中只 含有字符'0'
5	100000	100000	100000	
6				
7	1000	2000	2000	
8	100000	100000	100000	
9	300000	300000	1000000	
10				

红蓝树(math.cpp/math.in/math.out) 1s/1024M

问题描述

小 H 最近发现了一种数据结构“红蓝树”，这种树是一棵二叉树，并且对于所有非叶子节点：红色节点有两个蓝色的儿子节点，蓝色节点要么有两个蓝色的儿子节点，要么有恰好一个红色的儿子节点（要么为左儿子，要么为右儿子）

注意节点的儿子是有顺序的

令 f_n 表示有以蓝色节点为根的含有 n 个节点的树种类数。

小 H 并不关心 f_n 具体是多少，因为这个问题对他太简单了。为了考验你，它选择了 $2N$ 个数 $a_1, \dots, a_N, b_1, \dots, b_N$ ，并且计算了 $h_k = \sum_{i=1}^N a_i f^k(b_i)$ 。现在他告诉你 $b_1, \dots, b_N, h_1, \dots, h_N$ ，让你求 h_{N+1}

为了方便，所有的 h_i 对 998244353 取模

输入格式

第一行一个数 N

接下来一行 N 个数 b_1, \dots, b_N ,

接下来一行 N 个数 h_1, \dots, h_N

输出格式

一个数 h_{N+1}

样例输入

3

4 2 2

42 180 936

样例输出

5328

对应的 a 数组为：4 6 3

数据范围

测试点范围	$N \leq$	$b_i \leq$
1	1	10
2	100	
3		
4		
5	1000	1000
6		100000
7		
8	1000	1000
9		
10		
	60000	100000
		5000000