

NOIP 赛前集训营（第一场） 提高级

题目名称	语言	色球	斐波	偶数
题目类型	传统型	传统型	传统型	传统型
每个测试点 时限	C/C++ 1 秒, 其他语言 2 秒	C/C++ 1 秒, 其他语言 2 秒	C/C++ 1 秒, 其他语言 2 秒	C/C++ 2 秒, 其他语言 4 秒
内存限制	C/C++ 512MB, 其 他 语 言 1024MB	C/C++ 512MB, 其他语言 1024MB	C/C++ 512MB, 其 他 语 言 1024MB	C/C++ 512MB, 其他语言 1024MB
子任务数目	20	10	10	20
测试点是否 等分	是	是	是	是

语言 (language.cpp/in/out)

【题目描述】

Alice正在学习一种新的语言，在这种语言里，单词只有形容词(A)，名词(N)和动词(V)三种词性。但是一个单词可以对应多种词性。一个名词性词组(NP)可以由一个名词(N)，或者一个形容词修饰一个子名词性词组(A+NP₁)，或者两个子名词性词组(NP₁+NP₂)组成。即

$$NP ::= N \mid A + NP_1 \mid NP_1 + NP_2$$

一个句子(S)必须由一个名词性词组(NP₁)加一个动词(V)再加一个名词性词组(NP₂)组成，即

$$S ::= NP_1 + V + NP_2$$

Alice用这个语言写下一个单词的序列，现在你想知道这个单词序列是否能通过适当安程序列里每个单词的词性使之成为一个句子（不同位置的相同的单词也可以安排不同的词性）。

为了简单起见，她把每个单词编码对应为一个小写拉丁字母，不同的单词对应不同的字母（这里我们假设序列里面不同的单词的总数不超过 26 个）。每个单词用 1(001)到 7(111)来表示这个词性。数字的二进制第 1 位为 1 表示是 A（形容词），否则表示不是 A；第 2 位为 1 表示是 N（名词），否则表示不是 N；第 3 位为 1 表示是 V（动词），否则表示不是 V。

编码	词性
1	A
2	N
3	A or N
4	V
5	A or V
6	N or V
7	A or N or V

【输入格式】

第一行 T ，代表测试样例的个数。

对于每个测试样例，一行，26 个正整数 w_a, w_b, \dots, w_z ，表示每个单词的词性。

接下来一行，一个字符串 S ，表示单词序列，保证 S 中只包含小写字母。

【输出格式】

对于每组样例，如果能构成一个句子，输出“Yes”；否则输出“No”。

【样例 1 输入】

1

7 1 2 1

bcaa

【样例 1 输出】

Yes

【样例 1 说明】

首先 a 可以是 A 或 N 或 V；b 只能是 A；c 只能是 N。

当 b 是 A，c 是 N，bc 构成 NP；前面一个 a 是 V；后面一个 a 是 N，单独构成 NP，符号 NP+V+NP 的结构，因此可以是一个句子。

【样例 2 输入】

1

7 1 2 1

abca

【样例 2 输出】

No

【样例 2 说明】

可以验证，不管怎么给单词安排词性，都无法构成 NP+V+NP 的结构，所以不可能是一个句子。

【数据范围】

对于 30%数据，满足 $1 \leq |S| \leq 1000$

对于 100%数据，满足 $T = 10, 1 \leq |S| \leq 100000, 1 \leq wX \leq 7, X \in \{a, b, \dots, z\}$ 。

色球 (colorball.cpp/in/out)

【题目描述】

Bob有 n 种颜色的彩色小球 (编号 1 到 n)，每种颜色的小球他都有无限多个。他还有 n 个球桶 (编号 1 到 n)，球桶的内径与小球直径相当且高度是无限大的，因此能容纳无限多的小球。他想用这些小球和球桶玩游戏。

一开始这些球桶都是空的，紧接着他会顺次地做 m 个操作，每个操作都是以下 3 类操作中的一个。

1. 把 x 个颜色为 y 的彩色小球放到第 z 个桶的最上面 (*push x y z*) ;
2. 把最上面的 x 个小球从第 z 个桶内拿出来 (*pop x z*) ;
3. 把第 u 个桶的所有小球依次从顶部拿出放入第 v 个桶内 (*put u v*) 。

现在他已经确定好了这 m 个操作，但在他开始玩之前，他想知道每次他进行第二类操作取出的最后一个小球是什么颜色。

【输入格式】

第 1 行两个正整数 n, m 。

接下来 m 行每行是一个操作，

如果为第一类操作，则格式为*push x y z* (x, y, z 为正整数, $1 \leq x \leq 1e9, 1 \leq y, z \leq n$);

如果为第二类操作，则格式为*pop x z* (x, z 为正整数, $1 \leq x \leq 1e9, 1 \leq z \leq n$) 保证第 z 个桶内至少有 x 个小球;

如果为第三类操作，则格式为*put u v* (u, v 为正整数, $1 \leq u, v \leq n, u \neq v$)。

【输出格式】

对于每个第二类操作，输出Bob取出的最后一个小球的颜色编号。

【样例 1 输入】

2 5

push 1 1 1

push 1 2 1

push 2 3 2

put 1 2

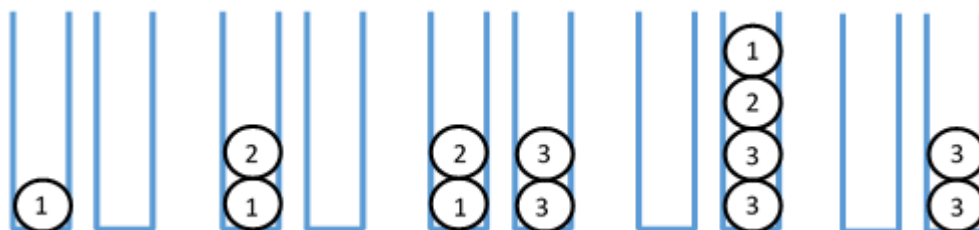
pop 2 2

【样例 1 输出】

2

【样例 1 说明】

5 个操作后的情况依次如下：



【数据范围】

对于 30% 的数据，满足 $1 \leq n, m \leq 1000$

对于另外 10% 的数据，所有第一类操作满足 $x = 1$

对于另外 10% 的数据，所有第二类操作满足 $x = 1$

对于 100% 的数据，满足 $1 \leq n, m \leq 200000$

斐波 (fib.cpp/in/out)

【题目描述】

假设 $fib(n)$ 为斐波那契数列的第 n 项, 其中 $fib(0) = 0, fib(1) = 1$, 且 $fib(n) = fib(n-1) + fib(n-2), (n > 1)$ 。

假设 S 是一个可重集合 $\{s_1, s_2, \dots, s_{|S|}\}$, $f(S)$ 定义为 $f(S) = \sum_{T \subseteq S} [fib(\sum_{s \in T} s)]^2$ 有

一个数组 a_1, a_2, \dots, a_n , Alice 会对数组进行 q 次操作, 每次操作可能是以下两种操作中的一种:

1. 把 a_p 变为 v ;
2. 计算 $\sum_{i=l}^r \sum_{j=i}^r f(\{a_i, a_{i+1}, \dots, a_j\})$ 。

对于每个操作 2, 输出答案模 998244353。

【输入格式】

第一行两个整数 n, q 。

第二行 n 个整数 a_1, a_2, \dots, a_n

接下来 q 行, 每行代表一个操作:

如果是操作 1, 格式为 $1 \ p \ v$;

如果是操作 2, 格式为 $2 \ l \ r$ 。

【输出格式】

输出答案, 模 998244353。

【样例 1 输入】

5 3

1 2 3 4 5

2 1 2

1 2 3

2 1 2

【样例 1 输出】

8

19

【样例 1 说明】

第一个询问:

$$f(\{1\}) = fib^2(0) + fib^2(1) = 1$$

$$f(1,2) = fib^2(0) + fib^2(1) + fib^2(2) + fib^2(3) = 6$$

$$f(\{2\}) = fib^2(0) + fib^2(2) = 1$$

$$f(\{1\}) + f(\{1,2\}) + f(\{2\}) = 8$$

第二个操作把数列改变为 1 3 3 4 5

第三个询问: $f(\{1\}) + f(\{1,3\}) + f(\{3\}) = 1 + 14 + 4 = 19$ 。

【样例 2 输入】

10 10

883 219 454 809 569 200 178 387 304 716

2 4 10

1 4 368

2 4 8

2 1 8

1 5 764

2 2 4

2 5 7

1 2 174

1 5 115

2 3 8

【样例 2 输出】

377148874

524944841

251771778

173440185

184756056

974115131

【数据范围】

对于 10%的数据，满足 $1 \leq n, q \leq 20$

对于 50%的数据，满足 $1 \leq n, q \leq 100$

对于 70%的数据，满足 $1 \leq n, q \leq 1000$

对于 100%的数据，满足 $1 \leq n, q \leq 100000$,

$1 \leq a_i \leq 10^5, \forall 1 \leq i \leq n$,

$1 \leq l \leq r \leq n, 1 \leq p \leq n, 1 \leq v \leq 10^5$ 。

偶数 (even.cpp/in/out)

【题目描述】

Bob喜欢偶数，他定义一种新的“偶数”为：(在十进制下，去掉前导零)数字的位数为偶数，且数字前一半和后一半完全一致。比如 121121、12341234 是“偶数”，而 111、121212 不是“偶数”。

对于一个“偶数”，Bob可以在这个“偶数”后继续添加数字，使得它成为新的“偶数”。比如，121121 可以在后面添加数字，使之变成 1211212112 成为新的“偶数”。Bob总是想添加最少的数字获得新的“偶数”。可以证明添加的方式是唯一的。

对于任何一个“偶数”，Bob都可以通过上述的方式产生新的“偶数”，这个新的“偶数”继续产生下一个新的“偶数”，直到这个“偶数”的位数超过任意给定的正整数 n 为止。之后，Bob会多次询问你，这个最终的“偶数”的第 l 位到第 r 位 ($1 \leq l \leq r \leq n$)组成的整数 模 998244353 后是多少。

【输入格式】

第一行 T ，表示测试样例数。

对于每个测试样例，一行一个“偶数” u 。（“偶数”的定义见题目描述）

接下来一行，两个正整数 n, q ，其中 n 表示Bob产生的最终的“偶数”的位数至少为 n ， q 代表询问个数。

接下来 q 行，每行两个正整数 l, r ，代表一个询问。

【输出格式】

对于每个询问输出答案，模 998244353。

【样例 1 输入】

2

121121

15 2

4 10

1 15

1111

10 2

1 5

1 10

【样例 1 输出】

1212112

230884310

11111

112866758

【样例 1 说明】

第 1 组：“偶数”产生直到超过 15 位：121121 → 1211212112 →
1211212112112121。

询问 1、4 到 10 位是 1212112。

询问 2、1 到 15 位是 121121211211212，取模后为 230884310。

【数据范围】

对于 10%的数据，满足 $1 \leq u \leq 10^{1000}$ ， $1 \leq n, q \leq 1000$

对于 40%的数据, 满足 $1 \leq n \leq 10^5$

对于 100%的数据, 满足 $T = 10, 1 \leq u \leq 10^{10^5}, 1 \leq n \leq 10^{18}, 1 \leq \sum q \leq 10^5,$
 $1 \leq l \leq r \leq n$ 。

保证 u 每一位只有 1-9。