

省选模拟赛

zzq

2019.03

题目名称	串	走	机
源程序文件名	str.c/cpp/pas	wlk.c/cpp/pas	cal1.out-cal10.out
输入文件名	str.in	wlk.in	N/A
输出文件名	str.out	wlk.out	N/A
时间限制	1s	3s	N/A
是否捆绑测试	是	是	是
内存限制	1GB	1GB	N/A
是否有部分分	否	否	是
题目类型	传统	传统	提交答案
是否有附加文件	否	否	否
编译开关	-O2 -std=c++11	-O2 -std=c++11	N/A

注意：

1. 评测时的栈空间大小为 256MB。
2. 今天的题目真的很简单，请大家不要假，认真做题。
3. 题目难度与顺序无关。
4. AK 了不要 D 出题人，没 AK 也不要 D 出题人。

Problem A. 串 (str.c/cpp/pas)

Input file: `str.in`
Output file: `str.out`
Time limit: 1 second
Memory limit: 1 gigabyte

zzq 喜欢做傻吊题。

一天，zzq 正在和往常一样做一个傻吊题：输入一个串，输出它共有多少个本质不同的子串。这里本质不同指的是长得不一样。由于这是一道 FJOI 题，字符只有 $\{A, C, G, T\}$ 四种。zzq 写了一个后缀数组，调了半小时通过了。zzq 感觉这个题没啥意思，应该加强一下。

记 $f(S)$ 为 S 的最小表示，具体地，把 S 中第一次出现的字母替换为 ' a '，第二次出现的字母替换为 ' b '，等等。例如， $f('AGACC') = 'abacc'$ 。题意仍然是输入一个串，输出它共有多少个本质不同的子串，但是现在两个串 A 和 B 本质不同定义为 $f(A)$ 与 $f(B)$ 不同。

由于现在这个题没那么傻吊了，zzq 不会做了。他想请你解决这个问题。

Input

一行一个由 $\{A, C, G, T\}$ 组成的非空字符串。

Output

一行表示答案。

Examples

str.in	str.out
ACGT	4
AAAA	4
ACGACGCAGTACAGACGACACGACGACACATC	383

Notes

设 n 为字符串的长度。对于所有数据， $n \leq 10^5$ 。

下表中留空表示无此限制。

子任务编号	子任务分值	$n \leq$	字符集
1	10	100	
2	20	1000	
3	20		$\{A, C\}$
4	30		$\{A, C, G\}$
5	20		

Problem B. 走 (wlk.c/cpp/pas)

Input file: wlk.in
Output file: wlk.out
Time limit: 1 second
Memory limit: 1 gigabyte

题目还是简单一点好。

有一张图，每一个点有至少一条出边，每条边上有一个小写字母。有一只大象在图上走路，一开始在 1 号点，之后每一步会随机选择该点的一条出边并走过去。

将大象走过的边上的小写字母顺次相连，就构成了一个字符串。大象有两个串 a 和 b ，如果大象走出的串包含 a 作为子串或包含 b 作为子序列，大象就会很生气，停止行走。

大象想知道在停止行走前大象期望会走几步。输入保证这个值是一个有理数，你只要输出它 mod 998244353 的值即可。

Input

第一行输入两个正整数 n, m ，表示点数和边数。

接下来 m 行，每行两个整数 x, y 和一个小写字母 c ，表示从 x 到 y 有一条有向边，边上的字符为 c 。

接下来两行每行一个由小写字母组成的非空字符串 a 和 b 。

保证两个相同的 x 和 y 之间不存在同样从 x 到 y ，边上字母相同的两条边。

Output

输出大象期望走的步数 mod 998244353 的值。

Examples

wlk.in	wlk.out
1 10 1 1 o 1 1 n 1 1 r 1 1 j 1 1 d 1 1 a 1 1 u 1 1 y 1 1 i 1 1 b doraadorab duanyaojiu	220364218
1 2 1 1 a 1 1 b aa aab	499122181
2 2 1 2 a 2 1 b abab aa	3

Notes

对于所有数据， $n \leq 20$, $|a| \leq 10$, $|b| \leq 50$ ，期望步数是一个确定的有理数。

下表中留空表示无此限制。

子任务编号	子任务分值	特殊条件
1	10	$n = 1$
2	20	每个点有恰好一条出边
2	20	$n \leq 5, b \leq 10$
2	50	

Problem C. 机 (cal1-10.out)

Input file: N/A
Output file: N/A
Time limit: N/A
Memory limit: N/A

这是一道提交答案题。

近期, ByteLand 出厂了一种新的计算机。

据可靠消息称, 这种计算机的设计者很喜欢乘法和快速幂, 不太喜欢加法和减法, 所以这台计算机只支持加常数。他还特别喜欢 10000121 这个数, 所以这台计算机内部只能存储 $[0, 10000120]$ 的整数, 并且运算都是在模 10000121 意义下的。这台计算机共有 10000121 个寄存器, 每个寄存器可以存储一个 $[0, 10000120]$ 的整数。

你刚刚拿到了一台最新出厂的该型号计算机, 你希望用它实现一些常见的计算任务。每一条指令需要相等的时间 (一个 cpu cycle) 来运行, 你希望最小化完成任务需要的时间。

0.1 语言说明

计算机内有 10000121 个寄存器, 在语言中可以用数组 $A[0..10000120]$ 访问。如前所述, A 的每一个位置可以存放一个 $[0, 10000120]$ 的整数。

我们称 $[0, 10000120]$ 的数为常数, 称 常数、 $A[\text{常数}]$ 和 $A[A[\text{常数}]]$ 为合法参数, 称一个长度不超过 15 的由小写字母组成的非空字符串为合法字符串。

以下为此计算机中的机器指令种类:

- 乘法: 接受合法参数 a , 合法参数 b 和合法参数 c , 令 $A[a] = b \times c \bmod 10000121$ 。(先计算右边, 再进行赋值, 下同)
- 快速幂: 接受合法参数 a , 合法参数 b 和合法参数 c , 令 $A[a] = b^c \bmod 10000121$ 。(定义 $x^0 = 1 (\forall x)$, $0^x = 0 (\forall x > 0)$)
- 加常数: 接受合法参数 a , 合法参数 b 和常数 c , 令 $A[a] = (b + c) \bmod 10000121$ 。
- 放置标签: 接受合法字符串 s , 把当前行置为标签 s 的跳转点。注意, 不能放置两个字符串一样的标签。
- 条件跳转: 接受合法参数 a , 合法参数 b 和合法字符串 s , 若 $a \neq b$ 则忽略本行, 否则跳转到标签 s 的跳转点 (需要保证存在 s 这一标签)。
- 输入: 接受合法参数 a , 从标准输入读入一个数, 存入 $A[a]$ 。
- 输出: 接受合法参数 a , 从标准输出输出 a 的值。

指令的调用格式:

- 乘法: mul a b c

- 快速幂: pow a b c
- 加常数: add a b c
- 放置标签: label s
- 条件跳转: if a b s
- 输入: input a
- 输出: output a

计算机顺序执行你给出的指令,直到执行到程序末尾,遇到条件跳转且满足条件时则跳转到对应行。

以下是一段示例程序,从标准输入读入两个数 a, b 。若 $ab \neq 0 \pmod{10000121}$, 输出 $(a+2) \pmod{10000121}$ 。

```
1 input 1
2 input 2
3 mul 3 A[1] A[2]
4 if A[3] 0 gg
5 add 3 A[1] 2
6 output A[3]
7 label gg
```

0.2 计算任务

共有 10 个计算任务,编号为 1 到 10。如无特殊注明,下面的输入均意为依次输入,输出均意为依次输出。

1. 输入一个 $[0, 10000120]$ 的整数 a , 输出 $a^{100000} \pmod{10000121}$ 。
2. 输入一个 $[1, 30000]$ 的整数 n 和 n 个 $[0, 10000120]$ 的整数 $a_1, a_2 \cdots a_n$, 倒序输出, 即依次输出 $a_n, a_{n-1}, a_{n-2} \cdots a_1$ 。
3. 输入两个 $[1, 20000]$ 的整数 n, q 和 n 个 $[1, 10000120]$ 的整数 $a_1, a_2 \cdots a_n$ 。接下来 q 次, 每次输入整数 l, r , 满足 $1 \leq l \leq r \leq n$, 你需要输出 $\prod_{i=l}^r a_i \pmod{10000121}$ 。
4. 输入一个 $[0, 10000120]$ 的整数 a , 若 a 为 2 的倍数输出 1, 否则输出 0。
5. 输入一个 $[0, 10000120]$ 的整数 a , 若 a 为 3 的倍数输出 1, 否则输出 0。
6. 输入两个 $[1, 5000000]$ 的整数 a 和 b , 输出 $a+b$ 。
7. 输入一个 $[1, 30000]$ 的整数 n 和 n 个正整数 $a_1, a_2 \cdots a_n$, 输出 $\sum_{i=1}^n a_i$, 保证这个值不超过 10000000。

8. 输入两个 $[1, 5000000]$ 的整数 a 和 b , 输出这两个数的最大公约数。
9. 输入一个 $[1, 2000]$ 的整数 a , 若 a 为质数输出 1, 否则输出 0。
10. 你需要提交一段有至少一个 **常数** 的代码, 它不接受输入, 依序输出了你提交的该段代码中的所有 **常数** 加一, 注意这里的加一不是模意义下的, 也就是说你的代码中不能包含 **常数** 10000120。例如上方的第一段示例程序代码中的 **常数** 依次为 $\{1, 2, 3, 1, 2, 3, 0, 3, 1, 2, 3\}$, 如果它的输出恰好为 $\{2, 3, 4, 2, 3, 4, 1, 4, 2, 3, 4\}$ 即合法。

0.3 评分规则

每个点你提交的代码中字符个数（包括空格和换行）不能超过 102400, 你可以粗略理解为每个点的输出文件大小不能超过 100kb。

你的代码会在一些输入上进行黑箱测试。如果你没能正确地完成计算任务（在一个或多个输入中返回错误的结果、代码格式错误等），得 0 分。否则你的得分将由在运行的每个输入中 $\max(\text{实际运行的除 label 外的行数}, \text{运行的 label 行数})$ 的最大值决定。

在选手目录下有与 10 个计算任务（即 10 个测试点）相对应的评分参数 cal1.ans-cal10.ans, 每个计算任务的评分参数为 10 个整数, 设你的代码的 $\max(\text{实际运行的除 label 外的行数}, \text{运行的 label 行数})$ 最大值为 c , 若 c 不超过其中的恰好 k 个整数, 你在该计算任务得 x 分。你在本题中的得分为每个点的得分之和。

0.4 附加提示

在选手目录下有 simulator.cpp, 编译后调用 simulator code.out <input.txt 即可测试代码 code.out 在输入 input.txt 上的输出（或返回错误信息）。spj 的实现与它基本相同, 如果发现 bug 请联系出题人。

你可以在代码中加入多余的空格或空行, 不过会算入代码字符限制（见评分规则第一段）。