

集合堆栈机

Setstack

【问题描述】

中学数学里集合的元素往往是具体的数字，比如 $A = \{1, 2, 3\}$ ， $B = \{\}$ （空集）等等。但是要特别注意，集合的元素也可以是另一个集合，比如说 $C = \{\{\}\}$ ，即说明 C 有且仅有一个元素——空集 B ，所以称 B 是 C 的子集或者称 B 是 C 的元素都是正确的。所谓一个集合的势，就是这个集合的元素个数，一般记为 $|S|$ ，空集的势为 0 。在上例中， $|A| = 3$ ， $|B| = 0$ ， $|C| = 1$ 。

鉴于集合论是现代数学的基础理论这一事实，一群异想天开的科学家开始着手建造一台新式的超级计算机——集合堆栈机 **Alpha**，这台机器操作的将是集合而不是数字。不过由于 **Alpha** 的竣工之日遥遥无期，科学家们希望你为他们编写一台虚拟机，好让他们检查自己的原型设计是否合理。

Alpha 的存储设备只有一个栈，栈的每个单元都只能放置一个集合。一开始，栈是空的，在每个操作结束后，计算机就会输出位于栈顶单元的那个集合的势。**Alpha** 拥有五种不同的指令，分别为：PUSH、DUP、UNION、INTERSECT 和 ADD，他们的功能如下：

- PUSH:** 把一个空集 $\{\}$ 压入栈；
- DUP:** 取出位于栈顶单元的集合，复制一遍以后再把两个同样的集合压入栈；
- UNION:** 取出位于栈顶单元的前两个集合，然后把它们的并集压入栈；
- INTERSECT:** 取出位于栈顶单元的前两个集合，然后把它们的交集压入栈；
- ADD:** 取出位于栈顶单元的前两个集合，首先取出的记为 S ，其次取出的记为 T ，最后把 $T \cup \{S\}$ 压入栈；

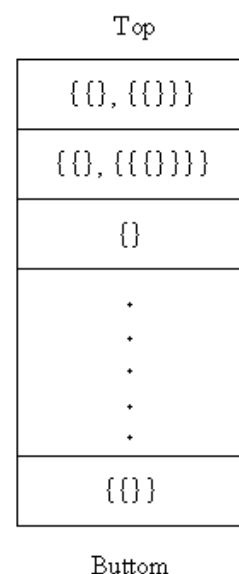
以左图为例，可见位于虚拟机堆栈顶端的两个元素是：

$$A = \{ \{\}, \{\{\}\} \}$$

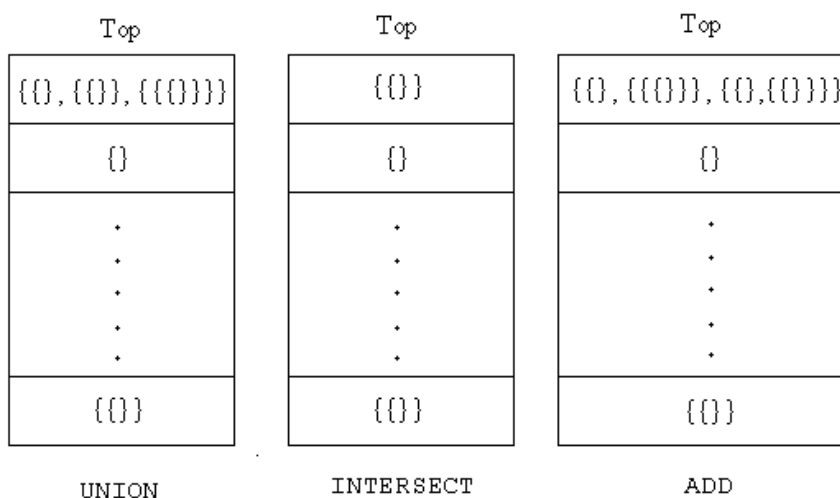
$$B = \{ \{\}, \{\{\{\}\}\} \}$$

根据势的定义，我们有 $|A| = 2$ 以及 $|B| = 2$ 。接下来，

- 如果选择 UNION 操作，结果是 $\{\{\}, \{\{\}\}, \{\{\{\}\}\}$ ，输出 3
- 如果选择 INTERSECT 操作，结果是 $\{\{\}\}$ ，输出 1
- 如果选择 ADD 操作，结果是 $\{\{\}, \{\{\{\}\}\}, \{\{\}, \{\{\}\}\}$ ，输出 3



分别执行三条指令之后虚拟机就会变成以下三种样子：



【输入文件】

文件的第一行只有一个整数 n ($0 \leq n \leq 2000$)，代表将要执行的指令条数。接下来有 n 行，每行有包含一条大写的指令，我们保证每条指令都是上述五条指令中的一条，并且虚拟机总是能正确执行完所有的指令。

【输出文件】

输出虚拟机的输出结果即可。每行输出一个大于或等于 0 的整数，代表虚拟机执行该条指令后的输出。选手们务必仔细考量程序的执行效率。

【样例】

setstack.in	setstack.out
9	0
PUSH	0
DUP	1
ADD	0
PUSH	1
ADD	1
DUP	2
ADD	2
DUP	2
UNION	
5	0
PUSH	0
PUSH	1
ADD	0
PUSH	0
INTERSECT	

【数据规模】

对于 20% 的数据， $n \leq 10$ 。

对于 100%的数据, $n \leq 2000$ 。