

2023 洛谷网校基础-提高衔接计划

模拟比赛

第三试

时间：2024 年 4 月 6 日 08:30 ~ 12:00

题目名称	吃药	字符串问题	分布式计算中心	序列问题
题目类型	传统型	传统型	传统型	传统型
目录	tablet	string	calc	sequence
可执行文件名	tablet	string	calc	sequence
输入文件名	tablet.in	string.in	calc.in	sequence.in
输出文件名	tablet.out	string.out	calc.out	sequence.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	20	20	20	25
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	tablet.cpp	string.cpp	calc.cpp	sequence.cpp
-----------	------------	------------	----------	--------------

编译选项

对于 C++ 语言	-O2 -std=c++14
-----------	----------------

注意事项

1. 文件名（程序名和输入输出文件名）必须使用英文小写；
2. C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须为 0。
3. 选手源程序必须存放于题目指定的目录（子文件夹）下。
4. 程序可使用的栈空间大小与该题内存空间限制一致。
5. 若无特殊说明，每道题的源代码大小限制为 100KB，输入与输出中同一行的相邻整数、字符串等均使用一个空格分隔，结果比较方式为全文比较（忽略行末空格、文末回车）。
6. 统一评测时采用的机器配置为：Intel(R) Core(TM) i5-12400 CPU @ 2.40GHz, RAM 16GB，上述时限以此配置为准。

7. 只提供 Linux 格式附加样例文件。
8. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。
9. 选手不得在源程序中使用内嵌汇编，也禁止以任何方式修改程序的编译选项。

吃药 (tablet)

【题目描述】

扶苏有两种药片，第一种药片每片含有效成分 a mg，第二种药片每片含有效成分 b mg。两种药片的有效成分不同，但是作用类似。具体而言， c mg 第一种药片的有效成分的作用等价于 d mg 第二种药片的有效成分。

药物有一定的副作用。具体来说，如果每日摄入超过 t mg 第一种药片的有效成分（如果摄入第二种药片，把摄入量换算成第一种药片的等价摄入量再计算），就会引发偏头痛等副作用。

现在，扶苏每天吃 x 片第一种药片加 y 片第二种药片。请你计算一下，她的药物摄入量是否会引发副作用。

【输入格式】

从文件 `tablet.in` 中读入数据。

本题单个测试点内有多组测试数据。输入的第一行是一个整数，表示数据组数 T 。

接下来 T 行，每行表示一组测试数据。每行只有七个整数，依次表示 a, b, c, d, t, x, y 。

【输出格式】

输出到文件 `tablet.out` 中。

对每组数据，依次输出答案。

如果会引发副作用，请你输出一行一个字符串 **Yes**。

否则请输出一行一个字符串 **No**。

【样例 1 输入】

```
1 2
2 1 1 2 1 4 2 1
3 1 2 357 272 6 1 2
```

【样例 1 输出】

```
1 No
2 Yes
```

【样例 1 解释】

对第一组数据, 两种药片都含 1 mg 有效成分, 2 mg 第一种药片的有效成分等价于 1 mg 第二种药片的有效成分。每天摄入第一种药片的有效成分不能超过 4 mg。扶苏每天吃 2 片第一种药片和 1 片第二种药片。吃一片第二种药片等于摄入了 2 mg 第一种药片的有效成分, 加上两片第一种药片也提供了 2 mg 有效成分, 共摄入了 4 mg 有效成分, 没有超过 4 mg, 因此不会引发副作用。

对第二组数据, 第一种药片含 1 mg 有效成分, 第二种药片含 2 mg 有效成分。每天吃 2 片第二种药片, 等于摄入 4 mg 第二种药片的有效成分, 也就是 5.25 mg 第一种药片的有效成分。加上吃一片第一种药片, 摄入了 1 mg 第一种药片的有效成分, 共摄入了 6.25 mg 第一种药片的有效成分, 超过了 6 mg, 会引发副作用。

【样例 2】

见选手目录下的 *tablet/tablet2.in* 与 *tablet/tablet2.ans*。

【子任务】

- 对 25% 的数据, $a = b, c = d$ 。
- 对 50% 的数据, $c = d$ 。
- 对 75% 的数据, $a, b, c, d, t, x, y \leq 100$ 。
- 另有 10% 的数据, $y = 0$ 。
- 对 100% 的数据, $1 \leq T \leq 2 \times 10^5, 1 \leq a, b, c, d \leq 10^6, 0 \leq x, y \leq 10^6, 1 \leq t \leq 3 \times 10^{18}$ 。

【提示】

请注意大量数据读入输出对程序效率造成的影响。

字符串问题 (string)

【题目背景】

在本题中，我们约定字符串和序列的下标都从 0 起，且用 $|A|$ 表示一个字符串 A 或一个序列 A 的长度。

【题目描述】

给定一个字符串 s 和一个字符串 t 。请你找到 s 的一个子序列使得这个子序列等于 t ，且每个字符出现在 s 中的位置尽可能靠前。

形式化地，你需要给出一个序列 p ，满足如下条件：

- p 的长度为 $|t|$ 。
- $0 \leq p_0 < p_1 < \cdots < p_{|t|-1} < |s|$ 。
- $s_{p_i} = t_i$
- 在所有符合要求的序列 p 中，你给出的序列字典序最小。

【输入格式】

从文件 *string.in* 中读入数据。

第一行是一个字符串，表示 s 。

第二行是一个字符串，表示 t 。

【输出格式】

输出到文件 *string.out* 中。

如果找不到符合要求的序列 p ，输出一行一个整数 -1 报告无解。

如果有解，为了避免输出过大，请输出一行一个整数，表示你找到的序列 p 的所有元素的按位异或和，即 $p_0 \oplus p_1 \oplus p_2 \oplus \cdots \oplus p_{|t|-1}$ 。这里 $\oplus$ 表示按位异或，即 C++ 语言中的 $\wedge$ 运算符。

【样例 1 输入】

```
1 aacbbbcc
2 abc
```

【样例 1 输出】

```
1 5
```

【样例 1 解释】

选出的序列 p 是 0,3,6，它们的按位异或和是 5。

【样例 2 输入】

```
1 acb
2 abc
```

【样例 2 输出】

```
1 -1
```

【样例 3】

见选手目录下的 *string/string3.in* 与 *string/string3.ans*。

【子任务】

- 对 15% 的数据，保证 $|s| = |t|$ 。
- 对 35% 的数据，保证 $|s| \leq 20$ 。
- 对 65% 的数据，保证 $|s| \leq 10^3$ 。
- 对 85% 的数据，保证 $|s| \leq 10^5$ 。
- 对 100% 的数据，保证 $1 \leq |t| \leq |s| \leq 10^7$ 。

分布式计算中心 (calc)

【题目背景】

扶苏在洛谷大厦建立了一个分布式计算中心。这个计算中心有若干个机房，每个机房里有若干台机器。每个机房和每个机器都有唯一编号。

有若干个程序在这个分布式计算中心运行，每个程序可能包含多个进程，每个进程一定属于且仅属于一个程序。每个进程可以在一台机器上运行，一台机器可以同时运行多个进程。

下图是一个关于机房、机器、程序、进程的简单示意图：

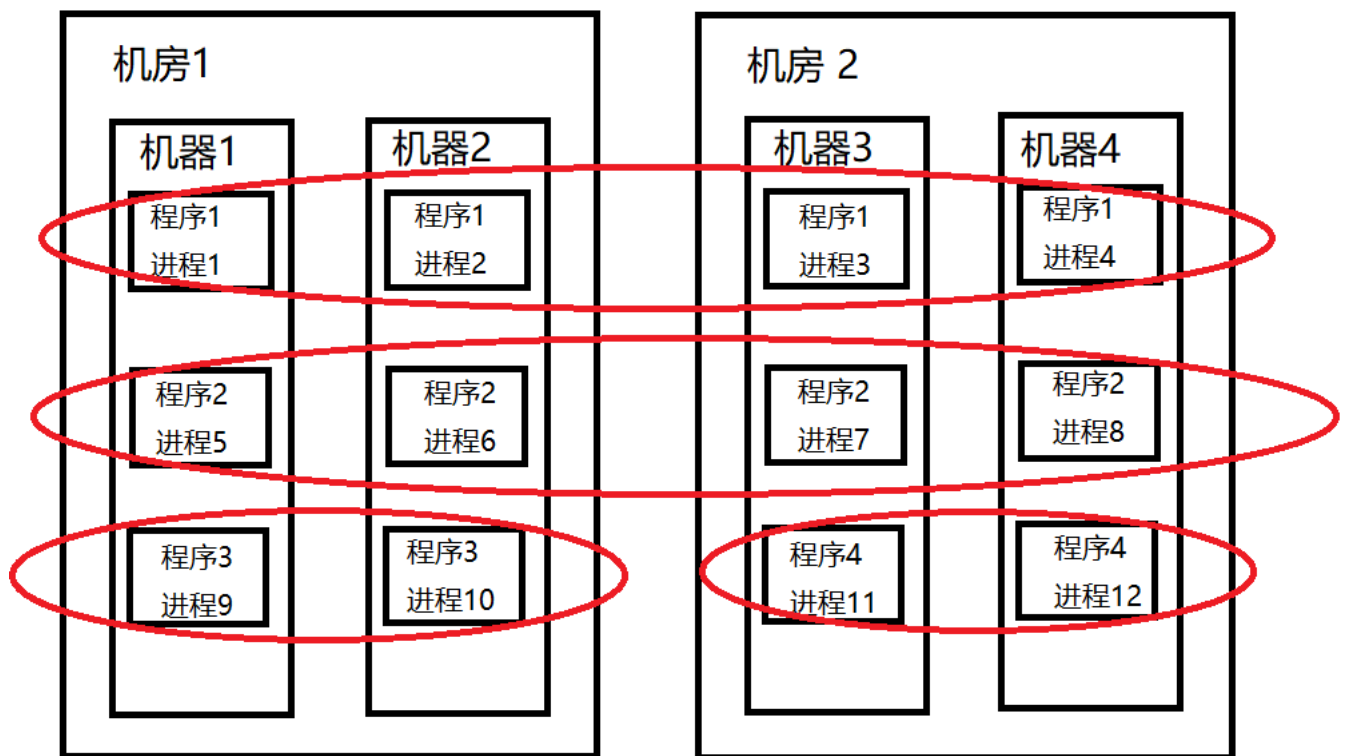


图 1: 示意图

【题目描述】

你，作为扶苏拉来的黑工，需要帮助扶苏接收各个程序的计算请求，并给他们分配相应的机房和机器。

处于安全考虑，每个进程都有自己的计算偏好，这些需求一共分为三类，每个进程都会有这三类偏好中的若干类：

1. 机房需求

这个进程必须在**指定的机房**内运行。

2. 正程序需求

这个进程必须和**指定的程序**在同一**机房**内运行。

这个要求对进程可以运行的**机房**进行了限制，在不考虑新加入的这个进程时，一个机房如果运行有指定的程序的进程（不一定要全部进程），则该进程可以在该机房运行。

3. 反程序需求

这个进程不能和**指定的程序**在同一**机器**上运行。

这个要求对进程可以运行的**机器**进行了限制，在不考虑新加入的这个进程时，一个机房如果运行有指定的程序的进程（不一定要全部进程），则该进程不能在该机器上运行。

当进程增多的时候，反程序需求可能很难被满足。因此在指定进程的反程序需求时，还要说明这个要求是“必须满足”还是“尽量满足”。

简单起见，在分配一个进程时，我们不考虑这个机器上已有进程的“反程序需求”。换句话说，即使这个机器上已经运行有一个进程且它的反程序需求要求不能和新被分配的进程在同一个机器上运行，我们同样可以把新进程分配到这个机器上，只要新进程的“反程序需求”被满足（或不能被满足但需求是“尽量满足”）。

扶苏要求你按照如下方式帮助她分配节点：按照进程到来的先后顺序，依次为每个进程分配对应的机器。一旦确定了一个进程对应的机器，就固定下来，且此后不在考虑这个进程的需求。

对每个进程，按如下两步分配机器：

1. 筛选阶段

在这个阶段，先根据进程的需求筛选出所有符合要求的机器。如果不存在符合要求的机器，且指定了“尽量满足”的反程序要求，则忽略反程序要求，再筛选一次。

如果筛选的结果是不存在符合要求的机器，则该进程分配失败，不给它分配机器。

2. 选择阶段

在所有被筛选出的机器中，选择此时运行的进程数最少的机器。如果有多个这样的机器，选择编号最小的。将该进程分配给该机器。

【输入格式】

从文件 `calc.in` 中读入数据。

第一行是两个整数，分别表示机器的数量 n 和机房的数量 m 。机器从 $1 \sim n$ 编号，机房从 $1 \sim m$ 编号。

第二行有 n 个整数，第 i 个整数表示 i 号机器所位于的机房 l_i 。

第三行有一个整数，表示到达的进程组数 q 。

接下来 q 行，每行六个整数 $f_i, a_i, na_i, pa_i, paa_i, paar_i$ ，依次表示到达的一组进程的信息，你需要依次为这些进程分配机器：

- f_i 表示到达了 f_i 个所属程序相同且需求相同的进程。
- a_i 表示这 f_i 个进程所属的程序编号。
- na_i 表示进程的机房需求：
当 $na_i = 0$ 时，表示没有机房需求，否则表示这些进程必须运行在编号为 na_i 的机房里。
- pa_i 表示正程序需求：
当 $pa_i = 0$ 时，表示没有正程序需求，否则表示这些进程必须和编号为 pa_i 的程序的某些进程运行在同一个机房内。
- paa_i 表示反程序需求：
当 $paa_i = 0$ 时，表示没有反程序需求，否则表示这些进程不能和编号为 pa_i 的程序的某些进程运行在同一个机器上。
- $paar_i$ 表示反程序需求是否必须满足：
如果 $paar_i = 0$ ，表示“尽量满足”；如果 $paar_i = 1$ ，表示“必须满足”。特别的，若 $paa_i = 0$ ，保证 $paar_i = 0$ 。

【输出格式】

输出到文件 `calc.out` 中。

共输出 q 行，每行 f_i 个整数。第 i 行的整数依次表示第 i 组到达的进程被分配的机器编号。如果某个进程分配失败，输出 0。

【样例 1 输入】

```

1 10 4
2 1 1 1 1 1 2 2 2 2 2
3 6
4 2 1 4 1 2 1
5 6 1 1 0 1 1
6 1 2 2 0 0 0
7 6 1 2 0 2 1
8 5 2 2 0 1 0
9 11 3 0 1 3 0

```

【样例 1 输出】

```

1 0 0
2 1 2 3 4 5 0
3 6

```

```

4 7 8 9 10 7 8
5 6 6 6 6 6
6 1 2 3 4 5 9 10 7 8 6 1

```

【样例 1 解释】

这个样例有 10 个机器，前五台机器属于机房 1，后五台机器属于机房 2。机房 3 和 4 没有机器。以下给出分配前两组进程的逻辑解释：

对第一组进程，它们的机房需求是 4，但机房 4 没有机器，故不能满足。

对第二组进程，它们的机房需求是 1。同时指定了必须满足的反进程需求：不能和应用 1 运行在同一台机器上。前五个进程可以依次被安排到前五台机器上。对第六个进程，它放在前五台机器上均不能满足反进程需求；放在其他机器上不能满足机房需求。因此它无法被分配。

【样例 2】

见选手目录下的 *calc/calc2.in* 与 *calc/calc2.ans*。

这个样例满足 $na_i = pa_i = paa_i = 0$ 。

【样例 3】

见选手目录下的 *calc/calc3.in* 与 *calc/calc3.ans*。

这个样例满足 $pa_i = paa_i = 0$ 。

【样例 4】

见选手目录下的 *calc/calc4.in* 与 *calc/calc4.ans*。

这个样例满足 $pa_i = 0$ 。

【样例 5】

见选手目录下的 *calc/calc5.in* 与 *calc/calc5.ans*。

【子任务】

以下用 A 表示 a_i 的上界。注意，我们不保证存在一个 i 使得 $a_i = A$ 。

对 100% 的数据，保证 $1 \leq m \leq n \leq 1000$, $1 \leq l_i \leq m$, $1 \leq q \leq \sum_{i=1}^q f_i \leq 4000$, $f_i \geq 1$, $1 \leq a_i \leq A \leq 10^9$, $0 \leq na_i \leq m$, $0 \leq pa_i, paa_i \leq A$, $0 \leq paar_i \leq 1$ 。

各测试点的信息如下表：

测试点编号	$A \leq$	$na_i = 0$	$pa_i = 0$	$paa_i = 0$
1,2	10	是	是	是
3,4	10^9			
5,6,7	10	否		否
8,9,10	10^9			
11,12,13	10			
14,15,16	10^9			
17,18	10			
19,20	10^9			

序列问题 (sequence)

【题目描述】

扶苏对数论颇感兴趣。她注意到当一个序列的最大公约数和最小公倍数可以很好地刻画这个序列的一些信息。

给定两个正常数 g, l ，如果一个长度为 n 的序列 a 满足：

1. 序列里的数都是正整数。
2. 序列里所有数字的最大公约数为 g 。
3. 序列里所有数字的最小公倍数为 l 。

则称这个序列是一个长度为 n 的正规序列。

现在，给定 n, g, l ，请你求出长度为 n 的正规序列的个数。扶苏知道这个答案可能很大，所以她只关心这个答案除以 998,244,353 的余数。

对两个长度为 n 的序列 a, b ，称 a 和 b 不同当且仅当存在一个 i 满足 $1 \leq i \leq n$ 且 $a_i \neq b_i$ 。

【输入格式】

从文件 `sequence.in` 中读入数据。

本题单个测试点内有多组测试数据。

输入的第一行是一个整数，表示测试数据组数 T 。

对每组测试数据，输入只有一行三个整数，分别表示 n, g, l 。

【输出格式】

输出到文件 `sequence.out` 中。

对每组测试数据，输出一行一个整数表示答案除以 998,244,353 的余数。

【样例 1 输入】

```
1 1
2 3 2 10
```

【样例 1 输出】

```
1 6
```

【样例 1 解释】

共有六种方案分别为：

- 2, 2, 10
- 2, 10, 10
- 2, 10, 2
- 10, 2, 2
- 10, 2, 10
- 10, 10, 2

【样例 2】

见选手目录下的 *sequence/sequence2.in* 与 *sequence/sequence2.ans*。

【子任务】

- 对 8% 的数据， $n = 2$ 。
- 对 28% 的数据， $g, l \leq 10$ 。
- 对 44% 的数据， $n \leq 5$ ， $g, l \leq 100$ 。
- 对 76% 的数据， $g, l \leq 100$ 。
- 对 100% 的数据， $1 \leq n \leq 5 \times 10^4$ ， $1 \leq g, l \leq 10^9$ ， $1 \leq T \leq 10$ 。