CSP2020 模拟赛

题目名称	DNA 序列	数列递推	七曜圣贤	旅游路线
题目类型	传统型	传统型	传统型	传统型
目录名	dna	seq	sage	trip
可执行文件名	dna	seq	sage	trip
输入文件名	dna.in	seq.in	sage.in	trip.in
输出文件名	dna.out	seq.out	sage.out	trip.out
每个测试点时限	1.0 sec	1.0 sec	2.5 sec	1.0 sec
内存限制	512 MiB	512 MiB	1024 MiB	512 MiB
测试点数量	10	20	10	20
单个测试点分值	10	5	10	5
结果比较方式	全文比较	全文比较	全文比较	全文比较
编译选项	-O2 -lm	-O2 -lm	-O2 -lm	-O2 -lm

DNA 序列 (dna)

【问题描述】

$CSP-S$ 复赛之前，HSD 桑进行了一项研究，发现人某条染色体上的一段 DNA 序列中连续的 k 个碱基组成的碱基序列与做题的 AC 率有关！于是他想研究一下这种关系。现在给出一段 DNA 序列，请帮他求出这段 DNA 序列中所有连续 k 个碱基形成的碱基序列中，出现最多的一种的出现次数。

【输入格式】

两行，第一行为一段 DNA 序列，保证 DNA 序列合法，即只含有 A, G, C, T 四种碱基；第二行为一个正整数，意义与题目描述相同。

【输出格式】

一行，一个正整数，为题目描述中所求答案。

【样例 1 输入】

AAAAA
1

【样例 1 输出】

5

【样例 1 解释】

对于这段 DNA 序列，连续的 1 个碱基组成的碱基序列只有 A，共出现 5 次，所以答案为 5。

【样例 2 输入】

ACTCACTC
4

【样例 2 输出】

2

【样例 2 解释】

对于这段 DNA 序列，连续的 4 个碱基组成的碱基序列为：$ACTC,CTCA,TCAC$ 与 $CACT$。其中 $ACTC$ 出现 2 次，其余均出现 1 次，所以出现最多的次数为 2，即为答案。

【数据规模与约定】

记 DNA 序列长度为 n。

下面给出每组数据的范围和满足性质情况：

<table>
<tr><th>测试点编号</th><th>n</th><th>k</th><th>其他</th></tr>
<tr><td>1</td><td>$=10^5$</td><td rowspan="2">$=1$</td><td>满足性质</td></tr>
<tr><td>$2,3$</td><td rowspan="2">$\le 5\times 10^5$</td><td></td></tr>
<tr><td>4</td><td rowspan="2">≤ 10</td><td>满足性质</td></tr>
<tr><td>$5\sim 8$</td><td>$\le 10^6$</td><td rowspan="2"></td></tr>
<tr><td>$9,10$</td><td>$=5\times 10^6$</td><td>$=10$</td></tr>
</table>

性质：给出的 DNA 碱基序列中每个碱基均相同。

对于所有数据均保证 $k\le n$

数列递推 (seq)

【问题描述】

$sosusosu$ 虐爆 OI 之后成为了一名文化课选手。一天，他做作业碰到了一堆数列问题，每道题给出的数列都是以下形式：

> 给定一个下标从 0 开始，无限长的整数列 $a_i, i \in \mathbb{N}$，已知 a_0, a_1 的值，
> 已经递推式 $a_{i+2} = ka_{i+1} + a_i, i \in \mathbb{N}, k \in \mathbb{N}^+$

$sosusosu$ 研究了这些数列，发现它们十分优美充满人类智慧，于是决定出一道 OI 题。

$sosusosu$ 给了你一个集合 $S \subset \mathbb{N}$，他想问你对于 S 中的每个数 s_i，使得 a_{s_i} 最大的 s_i 和使得 a_{s_i} 最小的 s_i 分别是多少。如果这样的 s_i 有多个，请你回答最小的一个。

另外，$sosusosu$ 准备对他作业中碰到的每个数列都让你回答一次，不过每次的集合 S 是一样的。

【输入格式】

输入第一行一个整数 m 表示 S 中元素个数。

第二行 m 个整 $s_1, s_2, ..., s_m$ 表示 S 中的元素。保证它们是非负整数且严格递增 (即 $s_i < s_{i+1}$)。

第三行一个整数 n 表示询问的数列个数。

接下来 n 行每行三个整数 a_0, a_1, k 描述一个数列。

【输出格式】

输出共 n 行，每行依次输出两个整数 $maxsi, minsi$，依次表示 S 的元素 s_i 中，使得 a_{s_i} 最大的 s_i 和使得 a_{s_i} 最小的 s_i 的值。如果这样的 s_i 有多个，请你回答最小的一个。

【样例 1 输入】

```
8
1 2 3 4 5 6 7 8
2
10 -6 1
0 0 1
```

【样例 1 输出】

```
2 1
1 1
```

【样例 1 解释】

第一个数列的前 9 项分别为 $10, -6, 4, -2, 2, 0, 2, 2, 4$，使得 a_{s_i} 最大的 s_i 为 2 和 $8(a_2 = a_8 = 4)$ 其中 2 较小，使得 a_{s_i} 最小的 s_i 为 $(a_1 = -6)$。第二个数列每项都等于 0，因此 S 中的每个元素 s_i 都既使 a_{s_i} 最大也使 a_{s_i} 最小，故答案是 S 中最小元素。

【样例 2 输入】

```
3
0 1 2
2
-2 3 1
3 -2 2
```

【样例 2 输出】

```
1 0
0 1
```

【样例 2 解释】

第一个数列的前 4 项分别为 $-2, 3, 1, 4$，使得 a_{s_i} 最大的 s_i 为 $1(a_1 = 3)$，使得 a_{s_i} 最小的 s_i 为 $0(a_0 = -2)$。第二个数列的前 4 项分别为 $3, -2, -1, -4$，使得 a_{s_i} 最大的 s_i 为 $0(a_0 = 3)$，使得 a_{s_i} 最小的 s_i 为 $1(a_1 = -2)$。

【样例 3】

样例 3 见下发文件 $seq3.in, seq3.ans$。

【更多的样例】

更多的样例见下发文件。其中除了前 3 个样例外还有约定分别和测试点 $1, 9, 14$ 相同的样例各一个。

【数据规模与约定】

对于所有数据，$1 \le n \le 3 \times 10^5, 1 \le m \le 10^5, 0 \le s_i \le 10^9, -10^7 \le a_0, a_1 \le 10^7, 1 \le k \le 5000$，保证 s_i 严格递增。

所有测试数据的范围和特点如下表所示：

(未标明的以上述所有数据的限制为准)

<table>
<tr><th>测试点编号</th><th>n,m 的限制</th><th>a_0,a_1,k 的范围</th><th>特殊限制</th></tr>
<tr><td>1</td><td rowspan="3">$n,m \le 100$</td><td rowspan="3">$-100 \le a_0,a_1 \le 100, k \le 10$</td><td rowspan="3">$s_m \le 10$</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td><td rowspan="12">$n \le 10^5$</td><td rowspan="3">$k=1$</td><td rowspan="3"></td></tr>
<tr><td>5</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td><td rowspan="5"></td><td>$a_0 \times a_1 \ge 0$</td></tr>
<tr><td>8</td><td>$|a_1| \ge |a_0|$</td></tr>
<tr><td>9</td><td rowspan="5">S 中的元素都是偶数</td></tr>
<tr><td>10</td></tr>
<tr><td>11</td></tr>
<tr><td>12</td><td>$k \le 10$</td></tr>
<tr><td>13</td><td>$k \le 100$</td></tr>
<tr><td>14</td><td>$k \le 1000$</td><td rowspan="7"></td></tr>
<tr><td>15</td><td></td></tr>
<tr><td>16</td><td>$n \le 1.5 \times 10^5$</td><td>$k \le 10$</td></tr>
<tr><td>17</td><td>$n \le 2 \times 10^5$</td><td>$k \le 100$</td></tr>
<tr><td>18</td><td>$n \le 2.5 \times 10^5$</td><td>$k \le 1000$</td></tr>
<tr><td>19</td><td colspan="2" rowspan="2"></td></tr>
<tr><td>20</td></tr>
</table>

七曜圣贤 (sage)

【问题描述】

不知道大家有没有听过物凄系列的一首歌，帕秋莉用卡车给博丽老板运货的故事。

又一次，卡车司机帕秋莉被拜托。红魔馆之主蕾米莉亚喜欢喝红茶，一天他要求帕秋莉开卡车帮他运红茶过来。

红茶其实是编好号了的，每个红茶都用一个非负整数来编号，从 0 开始一直到正无穷。帕秋莉请来好朋友魔理沙，帮他一起运红茶。

一开始卡车上已经有了编号为 0 到 a 的红茶（注意 $a=-1$ 就表示初始卡车上没有任何红茶），然后接下来到红魔馆的路上有 m 个时刻，每个时刻都会发生一种事件。

- 第一种事件，帕秋莉到了一个红茶店，买了一个编号为 x 的红茶 (卡车上初始没有这种编号的红茶，之前也不会买过相同编号的红茶)。
- 第二种事件，一个目前在卡车上的编号为 x 的红茶飞出了卡车。
- 第三种事件，魔理沙把目前不在卡车上的最早飞出去的红茶捡回了卡车上 (如果一个红茶曾经飞出去被捡回来过然后再飞出去，这里认为其飞出去的时间为最近一次飞出去的时间)。

由于描述这些事件实在是太麻烦了，聪明的魔理沙用了一个长度为 m 的整数序列 p 来描述每个时刻发生的事件。

- 这个序列 p 里所有元素均为 $[-1,b)$ 的整数。
- 若 $p_i=-1$ 则表示时刻 i 发生了第三种事件，如果此时并不存在满足条件的飞出去的红茶，则代表魔理沙脑子没转过来，忽视此次事件。
- 否则，如果在时刻 i 编号为 p_i 的红茶初始不在卡车上也从来没有通过第一种事件买过，则表示时刻 i 发生了一个买编号为 p_i 的红茶的第一种事件。
- 否则，如果在时刻 i 编号为 p_i 的红茶在卡车上，则表示时刻 i 发生了一个编号为 p_i 的红茶飞出卡车的第二种事件。
- 否则，表示时刻 i 发生了第三种事件，如果此时并不存在满足条件的飞出去的红茶，则忽视此次事件。

如果某个时刻的事件被忽视，那么我们**不执行对应的操作，也不计算此时的答案**。

帕秋莉是一个勤奋的人，每个时刻过后，如果这个时刻 i 发生了事件 (如果一个时刻发生的事件被忽视了，就不认为这个时刻发生了事件)，令 ans_i 表示时刻 i 过后卡车上所有编号小于 ans_i 的红茶都出现了，而编号为 i 的红茶没有出现 (很显然这个值是唯一的)。当然如果时刻 i 没有发生事件，则令 $ans_i = 0$。

请你对于 $1 \le i \le m$ 计算出 $ans_i \times (i^2 + 7i) \bmod 998244353$ 的异或和。

【输入格式】

第一行一个整数 T，表示数据组数。

接下来有 T 行，每行表示一组数据。

每组数据依次有 $m, seed, a, b, c, d$ 六个整数，其中 m, a, b 的意义与题面中相同；

d 表示是否只考虑第一种事件：d 的取值为 0 或 1，为特殊参数。当 $d = 1$ 时，请忽视所有的第二种事件与第三种事件 (忽视的含义见题面描述)。

$seed, c$ 是随机数生成器的参数。

我们使用如下实现的随机数生成器 $randnum()$。每组数据输入该组数据中 $seed$ 的初始值。

```
  unsigned 32bit integer seed

  function randnum()
    seed = seed xor (seed lsh 13)
    seed = seed xor (seed rsh 17)
    seed = seed xor (seed lsh 5)
    return seed
  end function
```

计算 $p[]$ 的代码如下：

```
for i = 1 to m by step 1
  if randnum() mod c == 0 then
    p[i] = -1
  else
    p[i] = randnum() mod b
  end if
end for
```

【输出格式】

每组数据输出一行表示答案。

【样例 1 输入】

1
7 327711436 4 6 3 0

【样例 1 输出】

292

【样例 1 解释】

p 序列为 $[5,-1,2,-1,2,5,4]$。初始时卡车上已经有了编号为 $[0,4]$ 的红茶。

第一个时刻，发生第一种事件，编号为 5 的红茶加入卡车，此时卡车上编号为 $[0,5]$ 的红茶都有，而编号为 6 的红茶没有，因此 $ans_1=6$。

第二个时刻，理论上应该发生第三种事件，但是并没有红茶飞出了卡车，因此该事件被忽视，$ans_2=0$。

第三个时刻，发生第二种事件，编号为 2 的红茶飞出卡车，此时卡车上编号为 $[0,1]$ 的红茶都有，而编号为 2 的红茶没有，因此 $ans_3=2$。

第四个时刻，发生第三种事件，魔理沙捡回编号为 2 的红茶回卡车，此时与第一个时刻后情况一致，因此 $ans_4=6$。

第五个时刻和第三个时刻一致，因此 $ans_5=2$。

第六个时刻，发生第二种事件，编号为 5 的红茶飞出卡车，此时卡车上编号为 $0,1,3,4$ 的红茶都有，而编号为 $2,5$ 的红茶没有，因此 $ans_6=2$。

第七个时刻，发生第二种事件，编号为 4 的红茶飞出卡车，此时卡车上编号为 $0,1,3$ 的红茶都有，而编号为 $2,4,5$ 的红茶没有，因此 $ans_7=2$。

【更多的样例】

更多的样例见下发文件。

【数据规模与约定】

对于所有数据 $1\le m\le 10^6, 1\le T\le 50, -1\le a\le m, 1\le b\le 2\times m, 1\le c\le 10^7, 0\le d\le 1$。

d 表示是否只考虑第一种事件，d 的取值为 0 或 1，为特殊参数。当 $d=1$ 时，请忽视所有的第二种事件与第三种事件 (忽视的含义见题面描述)。

注意，$d=1$ 时原本合法的事件也要被忽视，故即使你没有用到这个性质，也要记得判断 $d=1$ 的情况。除测试点 7 以外的测试点也有可能出现 $d=1$ 的数据。

<table>
<tr><th>测试点编号</th><th>m</th><th>T</th><th>特殊限制</th></tr>
<tr><td>1</td><td rowspan="3">≤ 3000</td><td>≤ 20</td><td rowspan="6">无</td></tr>
<tr><td>2</td><td>≤ 25</td></tr>
<tr><td>3</td><td>≤ 30</td></tr>
<tr><td>4</td><td rowspan="3">$m \le 10^5$</td><td>≤ 20</td></tr>
<tr><td>5</td><td>≤ 30</td></tr>
<tr><td>6</td><td rowspan="5">≤ 50</td></tr>
<tr><td>7</td><td>$\le 10^5$</td><td>$d=1$</td></tr>
<tr><td>8</td><td>$\le 8 \times 10^5$</td><td rowspan="3">无</td></tr>
<tr><td>9</td><td rowspan="2">$\le 10^6$</td></tr>
<tr><td>10</td></tr>
</table>

本题提供输入输出模板，在下发文件中 $sage$ 文件夹里有。

旅游路线 (trip)

【问题描述】

T 城是一个旅游城市，具有 n 个景点和 m 条道路，所有景点编号 $1, 2, ..., n$。每条道路连接这 n 个景区中的某两个景区，道路是**单向通行**的。每条道路都有一个长度。

为了方便旅游，每个景点都有一个加油站。第 i 个景点的加油站的费用为 p_i，加油量为 c_i。若汽车在第 i 个景点加油，则需要花费 p_i 元钱，之后车的油量将被加至**被加至**油量上限与 c_i 中的较小值。不过如果加油前汽车油量已经不小于 c_i，则不能在该景点加油。

小 C 准备来到 T 城旅游。他的汽车油量上限为 C。旅游开始时，汽车的油量为 0。在旅游过程中：

1、当汽车油量大于 0 时，汽车可以沿从当前景区出发的任意一条道路**到达**另一个景点 (不能只走道路的一部分)，汽车油量将减少 1；

2、当汽车在景点 i 且当前油量小于 c_i 时，汽车可以在当前景点加油，加油需花费 p_i 元钱，这样汽车油量将变为 $\min\{c_i, C\}$。

一次旅游的总花费等于每次加油的花费之和，旅游的总路程等于每次经过道路的长度之和。注意多次在同一景点加油，费用也要计算多次，同样地，多次经过同一条道路，路程也要计算多次。

小 C 计划旅游 T 次，每次旅游前，小 C 都指定了该次旅游的起点和目标路程。由于行程不同，每次出发前带的钱也不同。为了省钱，小 C 需要在旅游前先规划好旅游路线 (包括旅游的路径和加油的方案)，使得从起点出发，按照该旅游路线旅游结束后总路程不小于目标路程，且剩下的钱尽可能多。请你规划最优旅游路线，计算这 T 次旅游每次结束后最多可以剩下多少钱。

【输入格式】

输入第一行包含四个正整数 n, m, C, T，每两个整数之间用一个空格隔开，分别表示景点数、道路数、汽车油量上限和旅行次数。

接下来 n 行，每行包含两个正整数 p_i, c_i，每两个整数之间用一个空格隔开，按编号顺序依次表示编号为 $1, 2, ..., n$ 的景点的费用和油量。

接下来 m 行，每行包含三个正整数 a_i, b_i, l_i，每两个整数之间用一个空格隔开，表示一条从编号为 a_i 的景点到编号为 b_i 的景点的道路，道路的长度为 l_i。保证 $a_i \neq b_i$ ，但从一个景点到另一个景点可能有多条道路。

最后 T 行，每行包含三个正整数 s_i, q_i, d_i 描述一次旅游计划，旅游的起点为编号为 s_i 的景点，出发时带了 q_i 元钱，目标路程为 d_i。

【输出格式】

输出 T 行，每行一个整数，第 i 行的整数表示第 i 次旅游结束后最多剩下多少元钱。如果旅游无法完成，也就是说不存在从景点 s_i 出发用不超过 q_i 元钱经过不小于 d_i 的路程的路线，则该行输出 -1。

【样例 1 输入】

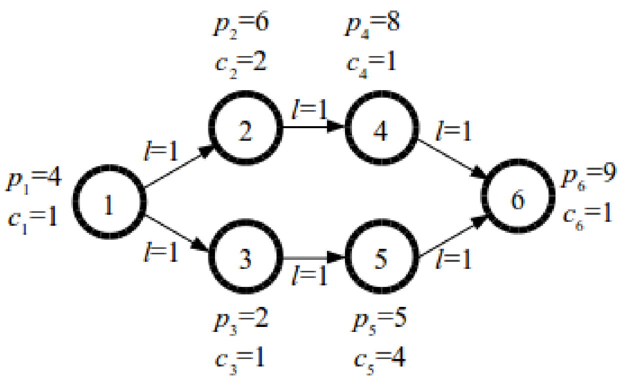
```
6 6 3 2
4 1
6 2
2 1
8 1
5 4
9 1
1 2 1
1 3 1
2 4 1
3 5 1
4 6 1
5 6 1
1 12 3
1 9 3
```

【样例 1 输出】

```
2
-1
```

【样例 1 解释】

T 城的景区和道路如下图所示：



由图可知，从景点 1 出发，路程为 3 的路线有两条：$1 \to 2 \to 4 \to 6$ 和 $1 \to 3 \to 5 \to 6$。

第 1 次旅游，最优路线为先在景点 1 加油，花费 4 元，此时油量为 1，然后到景 2，此时油量为 0，在景点 2 加油，花费 6 元，此时油量为 2，接着到景点 4，此时油量为 1，最后到景点 6，总路程为 3，最后剩余 $12 - 4 - 6 = 2$ 元。

第 2 次旅游，只用 9 元无论如何也无法走 3 的路程，因此旅游无法完成。

【样例 2 】

见下发文件的 $trip2.in$ 和 $trip2.ans$

【数据规模与约定】

<table>
<tr><th>测试点编号</th><th>n</th><th>m</th><th>C</th><th>T</th><th>p_i, c_i</th><th>特殊性质</th></tr>
<tr><td>1</td><td rowspan="4">≤ 10</td><td rowspan="4">$= n - 1$</td><td rowspan="7">≤ 10</td><td>$= 1$</td><td rowspan="7">≤ 10</td><td rowspan="4">$1, 2$</td></tr>
<tr><td>2</td><td rowspan="4">≤ 10</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td></tr>
<tr><td>5</td><td colspan="2">$= 10$</td><td rowspan="3">2</td></tr>
<tr><td>6</td><td colspan="2">$= 15$</td><td rowspan="2">≤ 20</td></tr>
<tr><td>7</td><td colspan="2">$= 20$</td></tr>
<tr><td>8</td><td rowspan="3">≤ 100</td><td rowspan="3">$= n - 1$</td><td rowspan="8">≤ 1000</td><td rowspan="8">≤ 50</td><td rowspan="5">≤ 100</td><td rowspan="3">$1, 3$</td></tr>
<tr><td>9</td></tr>
<tr><td>10</td></tr>
<tr><td>11</td><td rowspan="2">≤ 40</td><td rowspan="2">≤ 400</td><td rowspan="2">3</td></tr>
<tr><td>12</td></tr>
<tr><td>13</td><td rowspan="2">≤ 60</td><td rowspan="2">≤ 600</td><td rowspan="3">≤ 1000</td><td rowspan="8">无</td></tr>
<tr><td>14</td></tr>
<tr><td>15</td><td rowspan="2">≤ 80</td><td rowspan="2">≤ 800</td></tr>
<tr><td>16</td><td rowspan="5">$\le 10^5$</td><td rowspan="4">≤ 1000</td><td rowspan="5">$\le 10^5$</td></tr>
<tr><td>17</td><td rowspan="2">≤ 90</td><td rowspan="2">≤ 900</td></tr>
<tr><td>18</td></tr>
<tr><td>19</td><td rowspan="2">≤ 100</td><td rowspan="2">≤ 1000</td></tr>
<tr><td>20</td><td>$\le 10^5$</td></tr>
</table>