

第 37 届全国青少年信息学奥林匹克竞赛

CCF NOI 2020

第一试

时间：2020 年 8 月 18 日 08:00 ~ 13:00

题目名称	美食家	命运	时代的眼泪
题目类型	传统型	传统型	传统型
目录	delicacy	destiny	tears
可执行文件名	delicacy	destiny	tears
输入文件名	delicacy.in	destiny.in	tears.in
输出文件名	delicacy.out	destiny.out	tears.out
每个测试点时限	2.0 秒	2.0 秒	4.0 秒
内存限制	512 MB	512 MB	1 GB
子任务数目	20	25	25
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	delicacy.cpp	destiny.cpp	tears.cpp
-----------	---------------------	--------------------	------------------

编译选项

对于 C++ 语言	-lm -O2 -std=c++11
-----------	---------------------------

注意事项

1. 选手提交的源文件必须存放在已建立好的带有下发样例的文件夹中（该文件夹与试题同名）。
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 0。
4. 对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。
5. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格进行分隔。
6. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
7. 程序可使用的栈空间大小与该题内存空间限制一致。
8. 在终端下可使用命令 `ulimit -s unlimited` 将栈空间限制放大，但你使用的栈空间大小不应超过题目限制。

美食家 (delicacy)

【题目描述】

坐落在 Bzeroth 大陆上的精灵王国击退地灾军团的入侵后，经过十余年的休养生息，重新成为了一片欣欣向荣的乐土，吸引着八方游客。小 W 是一位游历过世界各地的著名美食家，现在也慕名来到了精灵王国。

精灵王国共有 n 座城市，城市从 1 到 n 编号，其中城市 i 的美食能为小 W 提供 c_i 的愉悦值。精灵王国的城市通过 m 条单向道路连接，道路从 1 到 m 编号，其中道路 i 的起点为城市 u_i ，终点为城市 v_i ，沿它通行需要花费 w_i 天。也就是说，若小 W 在第 d 天从城市 u_i 沿道路 i 通行，那么他会在第 $d + w_i$ 天到达城市 v_i 。

小 W 计划在精灵王国进行一场为期 T 天的旅行，更具体地：他会在第 0 天从城市 1 出发，经过 T 天的旅行，最终在恰好第 T 天回到城市 1 结束旅行。由于小 W 是一位美食家，每当他到达一座城市时（包括第 0 天和第 T 天的城市 1），他都会品尝该城市的美食并获得其所提供的愉悦值，若小 W 多次到达同一座城市，他将获得多次愉悦值。注意旅行途中小 W 不能在任意城市停留，即当他到达一座城市且还未结束旅行时，他当天必须立即从该城市出发前往其他城市。

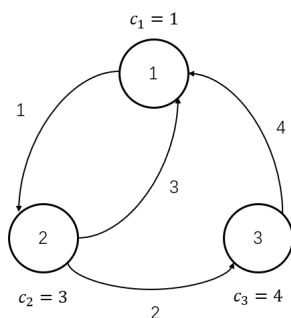


图 1: sample1

对于上图，小 W 一种为期 11 天的可行旅游方案为 $1 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 1$ ：

- 第 0 天，小 W 从城市 1 开始旅行，获得愉悦值 1 并向城市 2 出发。
- 第 1 天，小 W 到达城市 2，获得愉悦值 3 并向城市 1 出发。
- 第 4 天，小 W 到达城市 1，获得愉悦值 1 并向城市 2 出发。
- 第 5 天，小 W 到达城市 2，获得愉悦值 3 并向城市 3 出发。
- 第 7 天，小 W 到达城市 3，获得愉悦值 4 并向城市 1 出发。
- 第 11 天，小 W 到达城市 1，获得愉悦值 1 并结束旅行。
- 小 W 在该旅行中获得的愉悦值之和为 13。

此外，精灵王国会在不同的时间举办 k 次美食节。具体来说，第 i 次美食节将于第 t_i 天在城市 x_i 举办，若小 W 第 t_i 天时恰好也在城市 x_i ，那么他在品尝城市 x_i 的美食时会额外得到 y_i 的愉悦值。现在小 W 想请作为精灵王国接待使者的你帮他算出，他在旅行中能获得的愉悦值之和的最大值。

【输入格式】

从文件 *delicacy.in* 中读入数据。

第一行四个整数 n, m, T, k ，依次表示城市数、道路条数、旅行天数与美食节次数。

第二行 n 个整数 c_i ，表示每座城市的美食所能提供的愉悦值。

接下来 m 行每行三个整数 u_i, v_i, w_i ，依次表示每条道路的起点、终点与通行天数。

最后 k 行每行三个整数 t_i, x_i, y_i ，依次表示每次美食节的举办时间、举办城市与提供的额外愉悦值。

本题中数据保证：

- 对所有 $1 \leq i \leq m$ ，有 $u_i \neq v_i$ 。但数据中可能存在路线重复的单向道路，即可能存在 $1 \leq i < j \leq m$ ，使得 $u_i = u_j, v_i = v_j$ 。
- 对每座城市都满足：至少存在一条以该城市为起点的单向道路。
- 每次美食节的举办时间 t_i 互不相同。

【输出格式】

输出到文件 *delicacy.out* 中。

仅一行一个整数，表示小 W 通过旅行能获得的愉悦值之和的最大值。

若小 W 无法在第 T 天回到城市 1，则输出 -1。

【样例 1 输入】

```
1 3 4 11 0
2 1 3 4
3 1 2 1
4 2 1 3
5 2 3 2
6 3 1 4
```

【样例 1 输出】

```
1 13
```

【样例 1 解释】

该样例为题目描述中的例子，最优旅行方案见题目描述。

【样例 2 输入】

```
1 4 8 16 3
2 3 1 2 4
3 1 2 1
4 1 3 1
5 1 3 2
6 3 4 3
7 2 3 2
8 3 2 1
9 4 2 1
10 4 1 5
11 3 3 5
12 1 2 5
13 5 4 20
```

【样例 2 输出】

```
1 39
```

【样例 2 解释】

最优方案为 $1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1$ 。

- 第 0 天，小 W 从城市 1 开始旅行，获得愉悦值 3 并沿道路 3 通行。
- 第 2 天，小 W 到达城市 3，获得愉悦值 2 并沿道路 4 通行。
- 第 5 天，小 W 到达城市 4，由于美食节获得愉悦值 $20 + 4$ 并沿道路 7 通行。
- 第 6 天，小 W 到达城市 2，获得愉悦值 1 并沿道路 5 通行。
- 第 8 天，小 W 到达城市 3，获得愉悦值 2 并沿道路 4 通行。
- 第 11 天，小 W 到达城市 4，获得愉悦值 4 并沿道路 8 通行。
- 第 16 天，小 W 到达城市 1，获得愉悦值 3 并结束旅行。
- 小 W 获得的愉悦值之和为 39。

【样例 3】

见选手目录下的 *delicacy/delicacy3.in* 与 *delicacy/delicacy3.ans*。

该样例满足 $k = 0$ 。

【测试点约束】

对于所有测试点：

$1 \leq n \leq 50, n \leq m \leq 501, 0 \leq k \leq 200, 1 \leq t_i \leq T \leq 10^9。$

$1 \leq w_i \leq 5, 1 \leq c_i \leq 52501, 1 \leq u_i, v_i, x_i \leq n, 1 \leq y_i \leq 10^9。$

每个测试点的具体限制见下表：

测试点编号	n	m	T	特殊限制
1 ~ 4	≤ 5	≤ 50	≤ 5	无
5 ~ 8	≤ 50		≤ 52501	
9 ~ 10			$\leq 10^9$	A
11 ~ 13				$k = 0$
14 ~ 15				$k \leq 10$
16 ~ 17		无		
18 ~ 20	≤ 501			

特殊限制 A： $n = m$ 且 $u_i = i, v_i = (i \bmod n) + 1。$

命运 (destiny)

【题目描述】

提示：我们在题目描述的最后一段提供了一份简要的、形式化描述的题面。

在遥远的未来，物理学家终于发现了时间和因果的自然规律。即使在一个人出生前，我们也可以通过理论分析知晓他或她人生的一些信息，换言之，物理学允许我们从一定程度上“预言”一个人的“命运”。

简单来说，一个人的命运是一棵由时间点构成的有根树 T ：树的根结点代表着出生，而叶结点代表着死亡。每个非叶结点 u 都有一个或多个孩子 v_1, v_2, \dots, v_{c_u} ，表示这个人在 u 所代表的时间点做出的 c_u 个不同的选择可以导向的不同的可能性。形式化的，一个选择就是树上的一条边 (u, v_i) ，其中 u 是 v_i 的父结点。

一个人的一生是从出生（即根结点）到死亡（即某一个叶子结点）的一条不经过重复结点的路径，这条路径上任何一个包含至少一条边的子路径都是这个人的一段人生经历，而他或她以所有可能的方式度过一生，从而拥有的所有人生经历，都被称为潜在的人生经历。换言之，所有潜在的人生经历就是所有 u 到 v 的路径，满足 $u, v \in T$ ， $u \neq v$ ，并且 u 是 v 的祖先。在数学上，这样一个潜在的人生经历被记作有序对 (u, v) ，树 T 所有潜在的人生经历的集合记作 \mathcal{P}_T 。

物理理论不仅允许我们观测代表命运的树，还能让我们分析一些潜在的人生经历是否是“重要”的。一个人所作出的每一个选择——即树上的每一条边——都可能是重要或不重要的。一段潜在的人生经历被称为重要的，当且仅当其对应的路径上存在一条边是重要的。我们可以观测到一些潜在的人生经历是重要的：换言之，我们可以观测得到一个集合 $\mathcal{Q} \subseteq \mathcal{P}_T$ ，满足其中的所有潜在的人生经历 $(u, v) \in \mathcal{Q}$ 都是重要的。

树 T 的形态早已被计算确定，集合 \mathcal{Q} 也早已被观测得到，一个人命运的不确定性已经大大降低了。但不确定性仍然是巨大的——来计算一下吧，对于给定的树 T 和集合 \mathcal{Q} ，存在多少种不同的方案确定每条边是否是重要的，使之满足所观测到的 \mathcal{Q} 所对应的限制：即对于任意 $(u, v) \in \mathcal{Q}$ ，都存在一条 u 到 v 路径上的边被确定为重要的。

形式化的：给定一棵树 $T = (V, E)$ 和点对集合 $\mathcal{Q} \subseteq V \times V$ ，满足对于所有 $(u, v) \in \mathcal{Q}$ ，都有 $u \neq v$ ，并且 u 是 v 在树 T 上的祖先。其中 V 和 E 分别代表树 T 的结点集和边集。求有多少个不同的函数 $f: E \rightarrow \{0, 1\}$ （将每条边 $e \in E$ 的 $f(e)$ 值置为 0 或 1），满足对于任何 $(u, v) \in \mathcal{Q}$ ，都存在 u 到 v 路径上的一条边 e 使得 $f(e) = 1$ 。由于答案可能非常大，你只需要输出结果对 998,244,353（一个素数）取模的结果。

【输入格式】

从文件 `destiny.in` 中读入数据。

- 第一行包含一个正整数 n ，表示树 T 的大小，树上结点从 1 到 n 编号，1 号结点为根结点；

- 接下来 $n - 1$ 行每行包含空格隔开的两个数 x_i, y_i , 满足 $1 \leq x_i, y_i \leq n$, 表示树上的结点 x_i 和 y_i 之间存在一条边, 但并不保证这条边的方向;
 - 接下来一行包含一个非负整数 m , 表示所观测得到信息的条数。
 - 接下来 m 行每行包含空格隔开的两个数 u_i, v_i , 表示 $(u_i, v_i) \in \mathcal{Q}$ 。请注意: 输入数据可能包含重复的信息, 换言之可能存在 $i \neq j$, 满足 $u_i = u_j$ 且 $v_i = v_j$ 。
- 输入数据规模和限制参见本题末尾的表格。

【输出格式】

输出到文件 *destiny.out* 中。

- 输出仅一行一个整数, 表示方案数对 998,244,353 取模的结果。

【样例 1 输入】

```
1 5
2 1 2
3 2 3
4 3 4
5 3 5
6 2
7 1 3
8 2 5
```

【样例 1 输出】

```
1 10
```

【样例 1 解释】

共有 16 种方案, 其中不满足题意的方案有以下 6 种:

- $(1, 2), (2, 3), (3, 5)$ 确定为不重要, $(3, 4)$ 确定为重要: 集合 \mathcal{Q} 中没有限制被满足。
- $(1, 2), (2, 3), (3, 4), (3, 5)$ 确定为不重要: 集合 \mathcal{Q} 中没有限制被满足。
- $(1, 2), (2, 3)$ 确定为不重要, $(3, 4), (3, 5)$ 确定为重要: 集合 \mathcal{Q} 中 $(1, 3)$ 没被满足。
- $(1, 2), (2, 3), (3, 4)$ 确定为不重要, $(3, 5)$ 确定为重要: 集合 \mathcal{Q} 中 $(1, 3)$ 没被满足。
- $(2, 3), (3, 5)$ 确定为不重要, $(1, 2), (3, 4)$ 确定为重要: 集合 \mathcal{Q} 中 $(2, 5)$ 没被满足。
- $(2, 3), (3, 4), (3, 5)$ 确定为不重要, $(1, 2)$ 确定为重要: 集合 \mathcal{Q} 中 $(2, 5)$ 没被满足。
- 其他方案下, 集合 \mathcal{Q} 中的限制都被满足了。

【样例 2 输入】

```
1 15
2 2 1
3 3 1
4 4 3
5 5 2
6 6 3
7 7 6
8 8 4
9 9 5
10 10 7
11 11 5
12 12 10
13 13 3
14 14 9
15 15 8
16 6
17 3 12
18 5 11
19 2 5
20 3 13
21 8 15
22 1 13
```

【样例 2 输出】

```
1 960
```

【样例 3】

见选手目录下的 *destiny/destiny3.in* 与 *destiny/destiny3.ans*。

【样例 4】

见选手目录下的 *destiny/destiny4.in* 与 *destiny/destiny4.ans*。

【测试点约束】

测试点编号	n	m	T 为完全二叉树
1	≤ 10	≤ 10	否
2			
3			
4			
5	≤ 500	≤ 15	
6	≤ 10000	≤ 10	
7	≤ 100000	≤ 16	
8	≤ 500000		
9	≤ 100000	≤ 22	
10	≤ 500000		
11	≤ 600	≤ 600	
12	≤ 1000	≤ 1000	
13	≤ 2000	≤ 500000	
14			
15	≤ 500000	≤ 2000	
16			
17	≤ 100000	≤ 100000	是
18			
19	≤ 50000		
20	≤ 80000		
21	≤ 100000	≤ 500000	否
22			
23	≤ 500000		
24			
25			

全部数据满足： $n \leq 5 \times 10^5$, $m \leq 5 \times 10^5$ 。输入构成一棵树，并且对于 $1 \leq i \leq m$, u_i 始终为 v_i 的祖先结点。

完全二叉树：在本题中，每个非叶结点都有左右子结点，且所有叶子结点深度相同的树称为满二叉树；将满二叉树中的结点按照从上到下、从左向右的顺序编号，编号最小的若干个结点形成的树称为完全二叉树。

时代的眼泪 (tears)

【题目描述】

小 L 喜欢与智者交流讨论, 而智者也经常为小 L 出些思考题。

这天智者又为小 L 构思了一个问题。智者首先将时空抽象为了一个二维平面, 进而将一个事件抽象为该平面上的一点, 将一个时代抽象为该平面上一个矩形。

为了方便, 下面记 $(a, b) \leq (c, d)$ 表示平面上两个点 $(a, b), (c, d)$ 满足 $a \leq c, b \leq d$ 。

更具体地, 智者给定了 n 个事件, 他们用平面上 n 个不同的点 $\{(x_i, y_i)\}_{i=1}^n$ 来表示; 智者还给定了 m 个时代, 每个时代用平面上一个矩形 $(r_{i,1}, r_{i,2}, c_{i,1}, c_{i,2})$ 来表示, 其中 $(r_{i,1}, c_{i,1})$ 是矩形的左下角, $(r_{i,2}, c_{i,2})$ 是矩形的右上角, 保证 $(r_{i,1}, c_{i,1}) \leq (r_{i,2}, c_{i,2})$ 。我们称时代 i 包含了事件 j 当且仅当 $(r_{i,1}, c_{i,1}) \leq (x_j, y_j) \leq (r_{i,2}, c_{i,2})$ 。

智者认为若两个事件 i, j 满足 $(x_i, y_i) \leq (x_j, y_j)$, 则这两个事件形成了一次遗憾。而一个时代内包含的所有事件, 它们所形成的遗憾被称为这个时代的眼泪, 而形成的遗憾次数则称为该时代的眼泪的大小。现在智者想要小 L 计算每个时代的眼泪的大小。

小 L 明白, 如果他回答不了这个问题, 他也将成为时代的眼泪, 请你帮帮他。

【输入格式】

从文件 `tears.in` 中读入数据。

第一行两个整数 n, m , 分别表示事件数与时代数。

第二行 n 个整数 p_i , 其中第 i 个数表示事件 i 在平面上的坐标为 (i, p_i) 。保证 p_i 为一个 1 到 n 的排列。

之后 m 行, 每行四个整数 $r_{i,1}, r_{i,2}, c_{i,1}, c_{i,2}$, 表示每个时代对应的矩形。

【输出格式】

输出到文件 `tears.out` 中。

输出 m 行, 每行包含一个整数, 第 i 行输出第 i 个时代的眼泪的大小。

【样例 1 输入】

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

```
8 1 9 1 9
9 1 3 5 7
10 2 3 3 3
11 6 6 6 6
```

【样例 1 输出】

```
1 1
2 4
3 2
4 4
5 4
6 4
7 0
8 0
9 0
```

【样例 1 解释】

对于时代 1, 包含的遗憾有 (6, 7) (即事件 6 与事件 7 形成的遗憾, 下同)。

对于时代 2, 包含的遗憾有 (5, 6), (6, 7), (5, 7), (5, 8)。

对于时代 3, 包含的遗憾有 (5, 6), (5, 8)。

对于时代 4, 包含的遗憾有 (5, 6), (6, 7), (5, 7), (5, 8)。

对于时代 5, 包含的遗憾有 (5, 6), (6, 7), (5, 7), (5, 8)。

对于时代 6, 包含的遗憾有 (5, 6), (6, 7), (5, 7), (5, 8)。

对于时代 7, 8, 9, 它们均不包含任何遗憾。

【样例 2】

见选手目录下的 *tears/tears2.in* 与 *tears/tears2.ans*。

该样例满足特殊限制 A (具体限制见测试点约束)。

【样例 3】

见选手目录下的 *tears/tears3.in* 与 *tears/tears3.ans*。

该样例满足特殊限制 B (具体限制见测试点约束)。

【测试点约束】

对于所有测试点： $1 \leq n \leq 10^5$ ， $1 \leq m \leq 2 \times 10^5$ ， $1 \leq r_{i,1}, r_{i,2}, c_{i,1}, c_{i,2} \leq n$ 。

每个测试点的具体限制见下表：

测试点编号	$n \leq$	$m \leq$	特殊限制
1 ~ 3	10	10	无
4	3,000	3,000	
5	4,000	4,000	
6	5,000	5,000	
7	25,000	50,000	A
8	50,000	100,000	
9	75,000	150,000	
10	100,000	200,000	
11	60,000	120,000	B
12	80,000	160,000	
13	100,000	200,000	
14	20,000	40,000	无
15	30,000	60,000	
16	40,000	80,000	
17	50,000	100,000	
18	60,000	120,000	
19	70,000	140,000	
20 ~ 22	100,000	200,000	C
23 ~ 25			无

特殊限制 A：对于所有时代 i 有 $c_{i,1} = 1, c_{i,2} = n$ 。

特殊限制 B：任意两个不同时代所代表的矩形，它们要么是包含关系（一个矩形在另一个矩形内，边界允许重合），要么是相离关系（两矩形不包含共同点，边界不允许重合）。

特殊限制 C：最多有 50 对事件 (i, j) ($1 \leq i < j \leq n$) 不满足 $(i, p_i) \leq (j, p_j)$ 。