

# NOIP2024 模拟试题

时间：2024 年 11 月 26 日 08:00 ~ 12:30

题目名称	roadblock	artist	lone	station
题目类型	传统型	传统型	传统型	传统型
目录	roadblock	artist	lone	station
可执行文件名	roadblock	artist	lone	station
输入文件名	roadblock.in	artist.in	lone.in	station.in
输出文件名	roadblock.out	artist.out	lone.out	station.out
每个测试点时限	1.0 秒	2.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	256 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	roadblock.cpp	artist.cpp	lone.cpp	station.cpp
-----------	---------------	------------	----------	-------------

编译选项

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

### 注意事项（请仔细阅读）

- 文件名（程序名和输入输出文件名）必须使用英文小写。
- 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是 `0`。
- 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
- 选手提交的程序源文件必须不大于 100KB。
- 程序可使用的栈空间内存限制与题目的内存限制一致。
- 若无特殊说明，输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。
- 直接复制 PDF 题面中的多行样例，数据将带有行号，并且某些字符可能无法正常显示，建议选手直接使用对应目录下的样例文件进行测试。
- 评测时采用的机器配置为：Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz，内存 32GB。上述时限以此配置为准。

## roadblock

## 【题目描述】

小 X 是一名积木工程师，负责搭建一座宽度为  $n$  的大厦。大厦可以看成由  $n$  块宽度为 1 的积木组成，第  $i$  块积木的最终高度需要是  $h_i$ 。

在搭建开始之前，需要先填平下陷的地表。整段道路可以看作是  $n$  块相连的区域，首尾分别视作与  $-1$  与  $n+1$  这两个区域相连。一开始，第  $i$  块区域下陷的深度为  $d_i$ 。在这个阶段，小 X 每天可以选择一段连续区间  $[l, r]$ ，填充这段区间中的每块区域，让其下陷深度增加或减少 1。

小 X 不需要将整段道路的下陷深度都变为 0。她只需要保证任意相邻两个区域的下陷深度之差不超过  $k$ ，并且首尾两个区域下陷的深度绝对值也不超过  $k$ 。随后，来自五年之期的神秘力量就会帮她填平所有的地面。

接下来，就要开始搭建大厦了。每次操作，小 X 仍然可以选择一段连续区间  $[l, r]$ ，然后将第  $l$  块到第  $r$  块之间（含第  $l$  块和第  $r$  块）所有积木的高度分别增加 1。

小 X 是个聪明的小朋友，她很快想出了建造大厦的最佳策略，使得建造所需的操作次数最少。但她不是一个勤于动手的孩子，所以想请你帮忙实现这个策略，并求出最少的操作次数。

## 【输入格式】

从文件 `roadblock.in` 中读入数据。

包含两行，第一行包含两个整数  $n, k$ ，表示大厦的宽度（同时也是道路的长度），以及来自五年之期的神秘数字  $k$ 。

第二行包含  $n$  个整数，相邻两数间用一个空格隔开，第  $i$  个整数为  $d_i$ 。

第三行包含  $n$  个整数，第  $i$  个整数为  $h_i$ 。

## 【输出格式】

输出到文件 `roadblock.out` 中。

输出文件仅包含一个整数，即最少需要多少天才能完成任务。

## 【样例 1 输入】

```
1 5 1
2 4 3 2 5 3
3 2 3 4 1 2
```

【样例 1 输出】

110

【样例 1 解释】

在填平阶段，依次选择  $[1, 5](-1)$ ， $[1, 5](-1)$ ， $[4, 4](-1)$ ， $[4, 4](-1)$ ， $[1, 1](-1)$ ，最终结果为 1,1,0,1,1；

在建造阶段，依次选择  $[3, 5]$ ， $[1, 3]$ ， $[5, 5]$ ， $[1, 3]$ ， $[2, 3]$ 。

【样例 2】

见选手目录下的 *roadblock/roadblock2.in* 与 *roadblock/roadblock2.ans*。  
该样例数据满足测试点 7 ~ 12 的性质。

【样例 3】

见选手目录下的 *roadblock/roadblock3.in* 与 *roadblock/roadblock3.ans*。  
该样例数据满足测试点 13 ~ 16 的性质。

【样例 4】

见选手目录下的 *roadblock/roadblock4.in* 与 *roadblock/roadblock4.ans*。  
该样例数据满足测试点 17 ~ 20 的性质。

【数据范围】

对于 100% 的数据，满足  $1 \leq n \leq 100000, 0 \leq d_i, h_i, k \leq 10000$ 。

测试点编号	$n \leq$	特殊性质
1 ~ 6	10	无
7 ~ 12	1000	
13 ~ 16	$10^5$	A
17 ~ 20		无

特殊性质 A：满足  $k = 0$ ；

## artist

## 【题目背景】

吉娜是一名艺术家，一名表演艺术家（**performance artist**）。

此时正值虱子王国建国 **114514** 周年，虱子国王抵德请来艺术家吉娜设计游行的道路。

吉娜非常重视这份工作，但却在喝水的时候被巡逻的抵德训斥工作不认真，于是剥夺了吉娜的设计权利，将吉娜贬为了负责计算道路的信息的计算机。

吉娜心想：“为什么设计的时候不能喝水??? 我听歌都可以设计好。”

但终是敢怒不敢言，不幸的是，计算量实在太大，吉娜怕黑，必须赶在天黑之前做完工作。

于是他请来了你，伟大尼特，帮助它解决这个问题。

## 【题目描述】

给定一个长度为  $n$  的颜色序列  $c$ 。

再给出  $m$  个区间，第  $i$  个区间为  $[l_i, r_i]$ ，保证任何两个区间都是不相交或包含的关系。

在接下来的  $q$  个单位时间内，第  $i$  个时间会给定  $x, y$ ，表示将  $c_x$  变为  $y$ 。

请对于每一个区间求出，最早的其中所有颜色都互不相同的时间。

## 【输入格式】

从文件 **artist.in** 中读入数据。

第一行三个正整数  $n, m, q$ 。

接下来一行  $n$  个整数，第  $i$  个数表示第  $i$  个点的初始颜色  $c_i$ 。

接下来  $m$  行，每行两个整数  $l_i, r_i$ 。

接下来  $q$  行，每行两个整数  $x, y$ ，表示一次修改。

## 【输出格式】

输出到文件 **artist.out** 中。

令  $L_i$  表示第  $i$  个区间之中最早的所有颜色互不相同的时间，若在修改前就已经满足条件，则  $L_i = 0$ ，若不存在这样的时间，则令  $L_i = m + i$ 。

请输出  $\bigoplus_{i=1}^m L_i$ ，其中  $\bigoplus$  表示二进制下的按位异或。

## 【样例 1 输入】

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

**【样例 1 输出】**

```
1 4
```

**【样例 1 解释】**

$L_i$  依次为 7, 0, 0, 2, 4, 5。

**【样例 2】**

见选手目录下的 *artist/artist2.in* 与 *artist/artist2.ans*。

该样例满足测试点 1 ~ 2 的限制。

在选手目录下的 *artist/detailed\_artist2.ans* 中含有该样例不加密的答案。

**【样例 3】**

见选手目录下的 *artist/artist3.in* 与 *artist/artist3.ans*。

该样例满足测试点 6 ~ 8 的限制。

**【样例 4】**

见选手目录下的 *artist/artist4.in* 与 *artist/artist4.ans*。

该样例满足测试点 12 ~ 15 的限制。

**【样例 5】**

见选手目录下的 *artist/artist5.in* 与 *artist/artist5.ans*。

该样例满足测试点 18 ~ 20 的限制。

**【数据范围】**

对于所有数据，满足  $1 \leq n, m, q \leq 5 \times 10^5, 1 \leq l_i \leq r_i \leq n, 1 \leq c_i, y \leq n$ ，保证任何两个区间都是不相交或包含的关系，保证不存在两个完全相同的区间。

每个测试点的具体限制见下表：

测试点编号	$n, m, q \leq$	特殊性质
1 ~ 2	100	无
3 ~ 5	$5 \times 10^3$	
6 ~ 8	$10^5$	A
9 ~ 11	$5 \times 10^4$	B
12 ~ 15	$5 \times 10^5$	C
16 ~ 17	$10^5$	无
18 ~ 20	$5 \times 10^5$	

特殊性质 A：  $l_i = 1, r_i = i$ 。

特殊性质 B： 保证对于任意  $x$ ，满足  $l_i \leq x \leq r_i$  的  $i$  的数量不超过 40。

特殊性质 C： 保证初始时所有  $c_i$  相同，且每次修改的  $y$  与之前出现过的所有颜色不同。

## lone

## 【题目描述】

给你一棵树，你要去切这个树！

具体来说，最开始你有一棵树  $T_0$ 。

每次可以选择一条边  $(x, y)$  把这条边切断，然后选择剩下的两个连通块  $T_{i,1}, T_{i,2}$  其中一个，递归到子问题，即  $T_{i+1} \leftarrow T_{i,k}$ 。

你可以任意切  $k \leq 6$  次这棵树，但额外的，你希望每次切完树的大小是确定的，具体来说，给定  $k$  个正整数  $a_1, a_2, \dots, a_k$  你需要保证每次切树后， $|T_i| = a_i$ 。

请你计算有多少种本质不同的切树方法。答案对 998244353 取模。

我们记  $A = \{T_0, T_1, \dots, T_k\}$  表示每次切完剩下的树的集合，那么两种切树方案不同当且仅当，他们对应的  $A$  集合不同。

## 【输入格式】

从文件 ***lone.in*** 中读入数据。

第一行一个正整数  $n$  表示初始树的点数。

接下来  $n - 1$  行每行两个整数  $(u, v)$  描述一条树边。

接下来一行一个整数  $k$ ，含义同题目描述。

接下来一行  $k$  个整数分别表示  $a_1, a_2, \dots, a_k$ ，含义同题目描述。

## 【输出格式】

输出到文件 ***lone.out*** 中。

输出一行一个整数，表示合法切树的方案。

## 【样例 1 输入】

```
1 3
2 2 3
3 1 2
4 2
5 2 1
```

## 【样例 1 输出】

```
1 4
```

【样例 2】

见选手目录下的 *lone/lone2.in* 与 *lone/lone2.ans*。

【样例 3】

见选手目录下的 *lone/lone3.in* 与 *lone/lone3.ans*。

【样例 4】

见选手目录下的 *lone/lone4.in* 与 *lone/lone4.ans*。

【数据范围】

对于所有测试数据，保证  $2 \leq n \leq 5000, k \leq \min(n - 1, 6)$ ， $a_1 > a_2 > \cdots > a_k$ 。

测试点编号	$n \leq$	$k \leq$
1 ~ 3	5	5
4	100	2
5 ~ 6		3
7 ~ 8	5000	2
9 ~ 11		3
12 ~ 15		4
16 ~ 20		6



station

【题目描述】

$B$  市城内有  $n$  个车站,  $m$  条连接两个车站的有向线路, 现在小 C 随机选取  $k$  个车站, 作为“终点站”, 你要求出来原图的一个子图, 使得所有  $n$  个车站都可以到达某个“终点站”。

并且连接两个车站间的线路有长度, 你希望总长度和尽可能小。(也就是这个子图的边权和尽可能小)

输出最小的总长度和的期望长度。答案对 998244353 取模。

特别的, 如果存在一种选取车站的方式, 使得不存在合法选取子图的方式, 那么请只输出  $-1$ 。

特别注意, 图是有向图。

【输入格式】

从文件 `station.in` 中读入数据。

第一行三个非负整数  $n, m, k$  含义同题面。

接下来  $m$  行, 每行三个正整数  $u_i, v_i, w_i$  描述一条有向线路。

【输出格式】

输出到文件 `station.out` 中。

如果存在一种不合法方案, 请输出  $-1$ 。

否则输出一行一个整数, 表示答案对 998244353 取模的结果。

【样例 1 输入】

```
1 3 4 1
2 3 1 4
3 2 1 2
4 2 3 3
5 1 2 1
```

【样例 1 输出】

```
1 5
```

**【样例 2 输入】**

```
1 4 1 3
2 4 1 1
```

**【样例 2 输出】**

```
1 -1
```

**【样例 3】**

见选手目录下的 *station/station3.in* 与 *station/station3.ans*。  
该样例满足测试点 1 ~ 3 的限制。

**【样例 4】**

见选手目录下的 *station/station4.in* 与 *station/station4.ans*。  
该样例满足测试点 8 ~ 9 的限制。

**【样例 5】**

见选手目录下的 *station/station5.in* 与 *station/station5.ans*。  
该样例满足测试点 10 的限制。

**【样例 6】**

见选手目录下的 *station/station6.in* 与 *station/station6.ans*。  
该样例满足测试点 11 ~ 12 的限制。

**【样例 7】**

见选手目录下的 *station/station7.in* 与 *station/station7.ans*。  
该样例满足测试点 13 ~ 15 的限制。

**【样例 8】**

见选手目录下的 *station/station8.in* 与 *station/station8.ans*。  
该样例满足测试点 16 ~ 19 的限制。

【样例 9】

见选手目录下的 *station/station9.in* 与 *station/station9.ans*。  
该样例满足测试点 20 ~ 25 的限制。

【数据范围】

对于所有测试数据，保证  $2 \leq n \leq 10^5$ ， $1 \leq m \leq 2 \times 10^5$ ， $1 \leq k \leq n$ ， $0 \leq w_i \leq 998244352$ 。

测试点编号	$n \leq$	$m \leq$	特殊性质
1 ~ 3	15	30	无
4 ~ 7	20	50	
8 ~ 9	$10^5$	$2 \times 10^5$	A
10			B
11 ~ 12			C
13 ~ 15			D
16 ~ 19			E
20 ~ 25			无

- 特殊性质 A：满足  $w_i = 1$ ；  
特殊性质 B：满足  $k = n$ ；  
特殊性质 C：满足  $k = n - 1$ ；  
特殊性质 D：满足  $m = n$ ；  
特殊性质 E：满足  $k = 1$ 。