2025 NOIP 模拟赛

2025 NOIP 模拟赛

2025 年 9 月 30 日

时间：2025 年 9 月 30 日 08:00 ~ 12:00

题目名称	小猪盖房子	换乘旅行	优美的街景	网络规划
题目类型	传统型	传统型	传统型	传统型
目录	piggy	travel	street	network
可执行文件名	piggy	travel	street	network
输入文件名	piggy.in	travel.in	street.in	network.in
输出文件名	piggy.out	travel.out	street.out	network.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	10	10	10	10
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	piggy.cpp	travel.cpp	street.cpp	network.cpp

编译选项

对于 C++ 语言	-O2 -std=c++14 -static

小猪盖房子（piggy）

【题目描述】

猪大哥和猪小弟在成功抵御大灰狼后，决定建造一个更安全的家园。他们所在的区域是一块被划分为 n 行 m 列的巨大土地。

他们计划建造的房子有些特别。一栋房子会占据自上而下连续的若干（≥ 1）行，例如从第 u 行到第 d 行。在这些行的每一行中，都需要在**不同位置**建造两堵外墙，以围成一个内部区间。

土地上可能已经存在一些旧的墙体。对于第 i 行，可能的情况如下：

- **空地**：这一行没有任何旧墙体，小猪们可以在任意不相同的两列建造墙壁。
- **存在旧墙**：这一行已经存在两堵固定的墙，分别位于 c_1 列和 c_2 列。如果小猪们要将这一行纳入房子，那么他们**必须**使用这两堵旧墙，不能另起炉灶。

为了公平，他们决定建造两栋**完全一样**的房子。为了在危险来临时能相互照应，他们还计划让两栋房子**紧密相邻**。具体来说，如果第一栋房子占据了从第 u 行到第 d 行，那么第二栋房子就会占据紧随其后的、相同行数的区域，即从第 $d+1$ 行到第 $d+1+(d-u)$ 行。

两栋房子被称为“完全一样”，当且仅当对于任意 k（$0 \leq k \leq d-u$），第一栋房子在第 $u+k$ 行的墙体区间，与第二栋房子在第 $d+1+k$ 行的墙体区间完全相同（即两堵墙建造的位置相同）。

现在，小猪们想知道，在这片土地上，他们总共有多少种不同的盖房方案？一个方案由两栋房子占据的行（即 u 和 d）以及所有占据行上墙体的具体建造方式决定。

由于方案数可能非常大，请输出答案对 998244353 取模的结果。

【输入格式】

从文件 ***piggy.in*** 中读入数据。

第一行包含两个整数 n, m，分别表示土地的行数和列数。

接下来 n 行，每行包含两个整数 c_i, d_i，描述第 i 行的情况。

- 如果 $c_i = 0$ 且 $d_i = 0$，表示第 i 行是空地。
- 如果 $c_i > 0$，则表示第 i 行有旧墙，位置在 c_i 列和 d_i 列。保证 $1 \leq c_i < d_i \leq m$。

【输出格式】

输出到文件 ***piggy.out*** 中。

输出一行一个整数，表示答案对 998244353 取模的结果。

【样例 1 输入】

```
4 4
0 0
2 3
0 0
2 3
```

【样例 1 输出】

```
9
```

【样例 1 解释】

合法方案如下图所示，两个蓝色方框代表两栋房子占据的行，'#' 代表墙，其中红色的是新建的，黑色的是原有的。

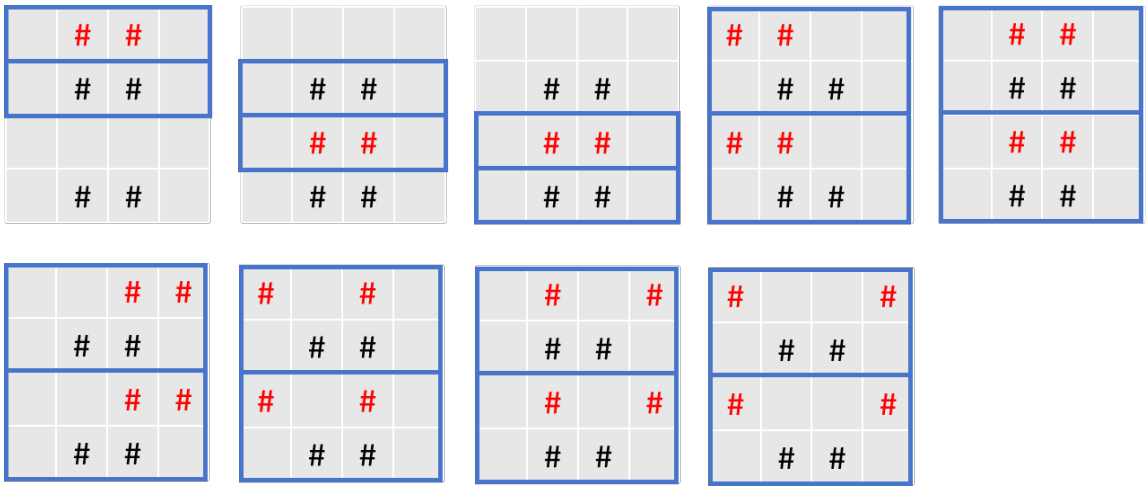

图 1: 样例一合法方案示意

【样例 2 输入】

```
10 6
0 0
2 3
0 0
1 4
0 0
2 3
```

```
0 0
1 4
0 0
4 5
```

【样例 2 输出】

```
465
```

【子任务】

对于所有数据，保证：

- $1 \leq n, m \leq 5000$；
- $1 \leq c_i < d_i \leq m$。

测试点	n, m
$1 \sim 5$	≤ 100
$6 \sim 7$	≤ 1000
$8 \sim 10$	≤ 5000

换乘旅行（travel）

【题目描述】

小明来到了一座著名的旅游城市，这座城市有一个包含 n 个站点的公共交通网络。

该网络的运行方式非常独特。每个站点 i 都有一个按顺序排列的摆渡车出发队列。每辆摆渡车都有一个固定的、预先设定的目的地站点。

一位旅客的行程如下：每当他到达一个站点（无论是起点还是中途换乘），他都必须搭乘该站点队列中最靠前的一辆摆渡车，这辆车会将他送到其目的地，然后不再运行。到达新站点后，他会继续按相同规则搭乘下一辆车。

当旅客到达一个没有任何待出发车辆的站点时，他的旅程就结束了，该站点即为他的最终目的地。

这些站点排成一排，从左到右依次编号为 1 到 n。小明想要知道，对于每一个站点，如果从它出发，最终会到达哪里。

【输入格式】

从文件 ***travel.in*** 中读入数据。

第一行包含一个整数 n，表示站点的数量。

接下来 n 行，第 i 行描述了第 i 个站点的摆渡车队列信息。

- 行首是一个整数 k_i，表示站点 i 的队列中有 k_i 辆摆渡车。
- 随后是 k_i 个整数 $d_{i,1}, d_{i,2}, \ldots, d_{i,k_i}$，**按出发顺序列出**了每辆车的目的地。$d_{i,1}$ 是第一辆出发的车的目标站，$d_{i,2}$ 是第二辆，以此类推。

【输出格式】

输出到文件 ***travel.out*** 中。

输出 n 行：

第 i 行包含一个整数，表示从站点 i 开始旅行的最终站点的编号。

【样例 1 输入】

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

【样例 1 输出】

```
1
2
2
```

【样例 1 解释】

以 1 号站台为例，从它开始旅行时，经过的站点为 1→1→2→3→1→2→1，最终目的地为 1。

【样例 2 输入】

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

【样例 2 输出】

```
1
1
1
1
1
```

【子任务】

对于所有数据，保证：

- $1 \le n \le 10^5$；
- $0 \le k_i \le 10^5$；
- $\sum_{i=1}^{n} k_i \le 10^6$；
- $1 \le d_{i,j} \le n$。

测试点	n	$\sum k_i$	附加限制
$1\sim 4$	≤ 1000	≤ 1000	无
$5\sim 7$	$\leq 10^5$	$\leq 10^5$	$\mathrm{k}_i = 1$
$8\sim 10$	$\leq 10^5$	$\leq 10^6$	无

优美的街景（street）

【题目描述】

在一条笔直的街道上，有 n 栋房屋一字排开，从左到右编号为 1 到 n。每栋房屋都有一个独一无二的高度，且房屋的高度恰好是 1 到 n 的一个排列。

一位城市规划师认为，一段连续的街景（即从第 l 栋到第 r 栋房屋，$1 \le l < r \le n$）是“优美”的，当且仅当这段房屋可以被分成**左右两个非空的部分**，并且左边所有房屋都严格矮于右边的所有房屋。

形式化地说，一段区间 $[l, r]$ 被称为“优美”的，当且仅当存在一个分割点 k（$l \le k < r$），使得 $\max(h_l, \ldots, h_k) < \min(h_{k+1}, \ldots, h_r)$。

例如，如果房屋高度序列为 $[3, 1, 4, 2]$，那么区间 $[3, 1, 4]$ 是优美的，因为我们可以在 1 和 4 之间分割，左半部分 $[3, 1]$ 的最大高度是 3，右半部分 $[4]$ 的最小高度是 4，满足 $3 < 4$。

现在，请你计算一下，在这条街道上，总共有多少个不同的区间 $[l, r]$ 是“优美”的？

【输入格式】

从文件 ***street.in*** 中读入数据。

第一行包含一个整数 n，表示房屋的数量。

第二行包含 n 个空格隔开的整数 $h_1, h_2, \ldots, h_n$，表示从左到右每栋房屋的高度。数据保证这是一个 1 到 n 的排列。

【输出格式】

输出到文件 ***street.out*** 中。

输出一行一个整数，表示“优美”区间的总数量。

【样例 1 输入】

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

【样例 1 输出】

```
7
```

【样例 1 解释】

优美的区间有：$[1, 4], [1, 5], [2, 4], [2, 5], [3, 4], [3, 5], [4, 5]$，共计 7 个。

【样例 2 输入】

```
6
1 6 2 4 3 5
```

【样例 2 输出】

```
10
```

【子任务】

对于所有数据，保证：

- $2 \leq n \leq 3 \cdot 10^5$；
- $h_1, \ldots, h_n$ 是 $1, \ldots, n$ 的一个排列。

测试点	n
$1 \sim 4$	≤ 100
$5 \sim 6$	≤ 3000
$7 \sim 10$	$\leq 3 \cdot 10^5$

网络规划（network）

【题目描述】

小 A 正在为一个由 n 台电脑组成的环形办公区设计网络方案。电脑按照环形排列，编号从 1 到 n。

最初的计划是，在所有相邻的电脑之间都铺设网线。特别地，连接电脑 i 和 $(i \pmod n) + 1$ 的网线成本为 c_i。然而，将所有 n 条网线都铺设的成本太高了。

就在小 A 苦恼时，他在公司的旧仓库里意外发现了一批闲置的集线器。这些集线器虽然有些老旧，但功能完好。根据说明书，每台集线器可以直连一个连续区间内的电脑，让它们之间可以相互连通。但是电脑数量不能超过 k 台，同时，每台电脑不能同时处于两台或以上的集线器的工作范围内，否则会导致冲突。

由于集线器本就是闲置的，因此无论使用多少集线器都不需要额外成本，小 A 决定结合集线器和网线这两种连接方式，设计一个更经济的方案。即他可以选择使用集线器将一段区间内的电脑直接相互连通，也可以选择使用网线将两台相邻的电脑连通，最终将所有电脑都连通在一起。

不过，网络设计的首要原则是稳定性。方案必须保证，在任意一条铺设的网线发生故障时，所有 n 台电脑之间仍然能够互相通信，你可以认为集线器不会出故障。

请你帮助小 A 找到一个满足上述要求的组网方案，使得总成本最低。

【输入格式】

从文件 ***network.in*** 中读入数据。

每个测试点包含多组测试数据。

第一行是一个整数 t ，表示测试数据的组数。

接下来包含 t 组数据，每组数据的格式如下：

- 第一行包含两个整数 n 和 k。
- 第二行包含 n 个整数 $c_1, c_2, \ldots, c_n$，其中 c_i 是连接电脑 i 和 $(i \pmod n) + 1$ 的网线成本。

【输出格式】

输出到文件 ***network.out*** 中。

对于每个测试用例，输出一行，包含一个整数，表示最小的总成本。

【样例 1 输入】

```
4
5 2
```

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

【样例 1 输出】

```
3
7
5
15
```

【样例 1 解释】

对于第一个样例，我们会把它分成 $\{1, 2\}, \{3, 4\}, \{5\}$，三组，每组内用集线器连接，组间（电脑 2-3, 电脑 4-5, 电脑 5-1）用网线连接。这样，总的成本为 $1 + 1 + 1 = 3$。

对于第二个样例，我们会把它分成 $\{5, 1\}, \{2\}, \{3, 4\}$，三组，每组内用集线器连接，组间（电脑 1-2, 电脑 2-3, 电脑 4-5）用网线连接。这样，总的成本为 $1 + 2 + 4 = 7$。

【子任务】

对于所有数据，保证：

- $1 \le t \le 10^5$；
- $2 \le n \le 5 \cdot 10^5$；
- $1 \le k \le n$；
- $1 \le c_i \le 10^9$;
- 一个测试点内 $\sum n \le 5 \cdot 10^5$。

测试点	$\sum n$	特殊性质
1∼ 4	≤ 2000	无
5	$\le 5 \cdot 10^5$	$c_i \le c_{i+1}$
6∼ 10	$\le 5 \cdot 10^5$	无