

NOIP 模拟赛 Round 3 -海亮高级中学

题目名称	购票方案	直线相交	记录数列	随机游走
目录	buy	intersect	record	walk
可执行文件名	buy	intersect	record	walk
输入文件名	buy.in	intersect.in	record.in	walk.in
输出文件名	buy.out	intersect.out	record.out	walk.out
每个测试点时限	1.0s	1.0s	1.0s	5.0s
内存限制	512MB	512MB	512MB	512MB
试题总分	100	100	100	100
测试点数目	20	20	20	20
每个测试点分值	5	5	5	5
是否有部分分	否	否	否	否
题目类型	传统型	传统型	传统型	传统型

提交的源程序文件名:

对于 C++ 语言	buy.cpp	intersect.cpp	record.cpp	walk.cpp
对于 C 语言	buy.c	intersect.c	record.c	walk.c
对于 Pascal 语言	buy.pas	intersect.pas	record.pas	walk.pas

编译开关:

对于 C++ 语言	-O2 -std=c++11	-O2 -std=c++11	-O2 -std=c++11	-O2 -std=c++11
对于 C 语言	-O2 -std=c11	-O2 -std=c11	-O2 -std=c11	-O2 -std=c11
对于 Pascal 语言	-O2	-O2	-O2	-O2

注意事项:

- 1、考试时长 4 小时;
- 2、每道题单独建子文件夹, 文件名(程序名和输入输出文件名)必须使用英文小写;
- 3、测试在 windows 环境下 lemonline 测试, 测试机 Intel Core i7-7700 3.60GHz。

1 购票方案 (buy)

1.1 题目描述

小 D 家附近新开了一家游乐园。小 D 经常光顾这家游乐园，他想要找到最优的购票方案。

小 D 想要进入游乐园 N 次，第 i 次在时刻 T_i ，这个游乐园有两种购票方式：

1. 购买单次票：价格为 One ，该票每张只能进入游乐园一次。

2. 购买时限票：时限票一共有 K 种类型，第 i 种类型的时限票可以使用的时长为 num_i ，价格为 $cost_i$ 。具体来说，若一张时限票在第 t 时刻被使用，则在 $[t, t + num_i - 1]$ 这个时刻区间内小 D 都可以进入游乐园。

这两种票都可以购买任意多张。记 sum_i 为小 D 前 i 次进入游乐园的最小花费（不考虑第 i 次之后的进入），小 D 定义第 i 次进入游乐园的代价为 $spend_i = sum_i - sum_{i-1}$ 。小 D 要知道每次进入游乐园的代价，即 $spend_1, \dots, spend_n$ 。

1.2 输入格式

从文件 buy.in 中读取数据。

第一行两个非负整数 N, K ，表示进入游乐园的总次数和时限票的种类数。

第二行 N 个正整数 T_1, \dots, T_N ，表示小 D 每次进入游乐园的时刻，保证单调增。

第三行到第 $K + 2$ 行，每行两个整数 $num_i, cost_i$ 表示每种时限票的时长和价格。

第 $K + 3$ 一个整数 One ，表示单次票的价格。

1.3 输出格式

输出到文件 buy.out 中。

N 行，每行一个整数 $Spend_i$ ，表示第 i 次进入游乐园的代价。

1.4 样例输入 1

```
3 2
10 20 30
90 50
1440 120
20
```

1.5 样例输出 1

```
20
```

20
10

1.6 样例解释

对于前 1 次进入，购买单次票，总价格为 20，故第 1 次进入的代价为 $20 - 0 = 20$ ；

对于前 2 次进入，购买两张单次票，总价格为 40，故第 2 次进入的代价为 $40 - 20 = 20$ ；

对于前 3 次进入，购买一张类型 1 时限票，总价格为 50，故第 3 次进入的代价为 $50 - 40 = 10$ 。

1.7 样例输入 2

```
10 2
13 45 46 60 103 115 126 150 256 516
90 45
450 100
20
```

1.8 样例输出 2

```
20
20
5
0
20
0
0
20
15
20
```

1.9 数据范围和约定

对于前 10% 的数据， $K = 0$ ；

对于前 40% 的数据， $K \leq 2, N \leq 10^3$ ；

对于前 60% 的数据， $K \leq 2, N \leq 10^5$ ；

对于前 80% 的数据， $K \leq 30, N \leq 10^5$ ；

对于 100% 的数据， $K \leq 500, N \leq 10^5, 1 \leq T_i, num_i, cost_i, One \leq 10^9, T_i$ 单调增。

2 直线相交 (intersect)

2.1 题目描述

游乐园内有这样一个项目：扔出 N 根木棒，猜这些木棒之间会有多少个交点。

形式化地，我们可以假设木棒的长度近似为无穷，木棒可以互相平行，但是不存在木棒相互重合或者三条及以上木棒交于同一点的情况。

小 D 想知道，如果扔出 N 根木棒，是否有可能出现总交点数为 M 的情况。

本题多组询问。

2.2 输入格式

从文件 `intersect.in` 中读取数据。

第一行一个整数 Q ，表示询问的数量。

第二行到第 $Q + 1$ 行，每行两个整数 N, M 表示一组询问：在扔出 N 根木棒的情况下是否有可能出现交点数为 M 的情况。

2.3 输出格式

输出到文件 `intersect.out` 中。

Q 行，每行一个整数 0 或者 1：1 表示有可能出现，0 表示不可能出现。

2.4 样例输入 1

```
4
3 0
3 1
3 3
4 4
```

2.5 样例输出 1

```
1
0
1
1
```

2.6 样例解释

下图分别是 3 根木棒摆出 0、3 个交点以及 4 根木棒摆出 4 个交点的情况。



2.7 样例输入 2

```
8
40 99
41 100
42 782
43 777
44 888
45 1000
46 1212
47 2333
```

2.8 样例输出 2

```
0
0
1
1
1
0
0
0
```

2.9 数据范围与约定

对于前 20% 的数据， $N \leq 5, M \leq 20$ ；

对于前 50% 的数据， $N \leq 10, M \leq 100$ ；

对于前 80% 的数据， $N \leq 50, M \leq 2500, Q \leq 10$ ；

对于 100% 的数据， $N \leq 500, 0 \leq M \leq 10^5, Q \leq 10^5$ 。

3 记录数列 (record)

3.1 题目描述

对于一个 $1 - N$ 的排列 p ，小 D 如下定义长度为 N 的阶梯数组 a ：

a_i 为包含下标 i 的由单调且连续的元素构成的区间的最大长度，即在 p 中这个区间要么是 $[x, x+1, x+2, \dots, y-1, y]$ 或者是 $[y, y-1, \dots, x+1, x]$ 的形式。

例如，对于排列 $p = [4, 1, 2, 3, 7, 6, 5]$ ，我们有 $a = [1, 3, 3, 3, 3, 3, 3]$ 。

现在，小 D 已经知道了阶梯数组 a ，他想知道有多少个排列的阶梯数组等于 a ，由于答案可能很大，小 D 只想知道答案对 998244353 取模后的结果。

3.2 输入格式

从文件 record.in 中读取数据。

第一行一个整数 N ，表示阶梯数组的长度。

第二行 N 个正整数 a_1, \dots, a_N ，描述阶梯数组。

3.3 输出格式

输出到文件 record.out 中。

一行一个整数 Ans ，表示排列的数量对 998244353 取模后的结果。

3.4 样例输入 1

```
6
3 3 3 1 1 1
```

3.5 样例输出 1

```
6
```

3.6 样例解释

可能的排列 p 如下：

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

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

3.7 样例输入 2

```
8
2 2 2 2 2 2 1 1
```

3.8 样例输出 2

```
370
```

3.9 数据范围与约定

对于前 20% 的数据, $N \leq 8$;

对于前 50% 的数据, $N \leq 16$;

对于另外 10% 的数据, 保证 $a_i = 1$;

对于另外 20% 的数据, 保证 $a_i > 1$;

对于 100% 的数据, $N \leq 2000$, $1 \leq a_i \leq N$ 。

4 随机游走 (walk)

4.1 题目描述

给定一个 N 个点 M 条边的简单无向连通图，第 i 条边上有边权 w_i ，小 D 将在这个无向图中进行随机游走。

具体来说：在第 0 时刻，小 D 可能出现在 $1, 2, \dots, N-1$ 号节点中的任意一个节点上，第 i 个节点的出现权重为 a_i （即小 D 出现在第 i 个节点的概率 $p_i = \frac{a_i}{\sum_{j=1}^{N-1} a_j}$ ），小 D 不会出现在 N 号节点。

在每个时刻，小 D 会等概率地随机选择当前所在节点的一条出边 j ，走向这条出边对应的节点并获得边权 w_j 的分数。若在某一时刻，小 D 到达了 N 号节点，随机游走立刻结束。

该无向图会进行 Q 次修改，每次修改为以下两种修改中的一种：

1. 改变某条边的权重；
2. 改变某个点的出现权重。

修改是长久的，即第 $i+1$ 次修改基于第 i 的修改后得到的无向图。

小 D 想知道，在第一次修改之前和每次修改之后，小 D 在这张无向图上进行随机游走期望获得的分数，对 998244353 取模。

4.2 输入格式

从文件 walk.in 中读取数据。

第一行三个整数 N, M, Q ，表示无向图的点数，边数和对无向图的修改数。

第二行 $N-1$ 个整数 a_1, \dots, a_{N-1} ，描述出现权重数列 a 。

第三行到第 $M+2$ 行，每行三个整数 u_i, v_i, w_i ，描述无向图中的一条边。

第 $M+3$ 行到第 $Q+2$ 行，每行描述一次修改，具体如下：

首先读入一个数 opt_i ，描述修改的种类。

若 $opt_i = 1$ ，随后会读入 3 个整数 u_i, v_i, w_i ，表示将 (u_i, v_i) 这条边的边权改为 w_i 。（保证 (u_i, v_i) 这条边存在）

若 $opt_i = 2$ ，随后会读入 2 个整数 id_i, a_i ，表示将第 id_i 个点的出现权重改为 a_i ，保证 $1 \leq id_i \leq N-1$ 。（注意在该修改后所有点的出现概率都有可能改变）

4.3 输出格式

输出到文件 walk.out 中。

输出共 $Q+1$ 行。

第一行一个整数，描述未修改时随机游走的期望分数。

接下来 Q 行，每行一个整数，描述第 i 次修改后随机游走的期望分数。

4.4 样例输入 1

```
3 2 4
1 0
1 2 0
2 3 1
1 1 2 1
2 2 1
1 2 3 0
2 1 0
```

4.5 样例输出 1

```
1
4
499122180
499122179
2
```

4.6 样例解释

当小 D 位于 1 号节点时，他下一步一定会走向 2 号节点，当小 D 位于 2 号节点时，他下一步有 $1/2$ 的概率会走向 1 号节点，有 $1/2$ 的概率会走向 3 号节点，当小 D 位于 3 号节点时，游走结束。

若小 D 从 2 节点开始游走，他期望经过 1 号节点的次数为 1，经过 1 - 2 这条边的期望次数为 2，2 - 3 这条边的期望次数为 1；

若小 D 从 1 节点开始游走，他期望再次经过 1 号节点的次数为 1，经过 1 - 2 这条边的期望次数为 3，2 - 3 这条边的期望次数为 1。

在修改未发生之前，只有 2 - 3 号边上有权值 1，故期望分数为 1；

在第一次修改发生后，期望分数为 $1 \times 3 + 1 \times 1 = 4$ ；

在第二次修改发生后，期望分数为 $1 \times (0.5 \times 3 + 0.5 \times 2) + 1 \times 1 = 3.5$ ；

在第三次修改发生后，期望分数为 $1 \times (0.5 \times 3 + 0.5 \times 2) + 0 \times 1 = 2.5$ ；

在第四次修改发生后，期望分数为 $1 \times 2 + 0 \times 1 = 3$ 。

4.7 样例输入 2

```
6 8 9
```

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

4.8 样例输出 2

```
146800688
88080431
639500335
352321583
528482348
391468415
313174743
821471960
779878448
800675202
```

4.9 数据范围与约定

对于前 30% 的数据，保证给定的无向图为 $1 \rightarrow 2 \rightarrow \dots \rightarrow N$ 的一条链；

对于前 50% 的数据， $N \leq 20, M \leq 30, Q \leq 5$ ；

对于另外 10% 的数据， $Q = 0$ ；

对于另外 15% 的数据，保证不会出现种类为 2 的修改操作；

对于另外 15% 的数据，保证只有连向 N 节点的边的边权不为 0；

对于 100% 的数据， $N \leq 500, M \leq 10^4, q \leq 10^6, 0 \leq w_i, a_i \leq 10^4$ ，保证种类为 2 的操作最多出现 N 次。