

## 2023 牛客 OI 赛前集训营-提高组（第二场）

比赛地址: <https://ac.nowcoder.com/acm/contest/65193>

题目名称	集合	出租	连通块	跳棋
题目类型	传统型	传统型	传统型	传统型
每个测试点时限	C/C++1 秒, 其他语言 2 秒	C/C++1 秒, 其他语言 2 秒	C/C++1 秒, 其他语言 2 秒	C/C++1 秒, 其他语言 2 秒
内存限制	C/C++256MB, 其他语言 512MB	C/C++256MB, 其他语言 512MB	C/C++256MB, 其他语言 512MB	C/C++256MB, 其他语言 512MB
子任务数目	10	20	20	20
测试点是否等分	是	是	是	是

### 注意事项

- 所有参与牛客 OI 赛前集训营的选手必须遵守约定的纪律:
    - 比赛账号不能外传。
    - 比赛中不能抄袭代码。
    - 比赛中不能恶意卡评测。
  - 报名支付账号即为比赛账号。
  - 一旦报名牛客 OI 赛前集训营活动, 不支持退费, 请考虑清楚后报名。
  - 本活动解释权归牛客网所有, 活动介绍未尽事宜以牛客网官方解释为准。
- 欢迎关注“比赛自动姬”公众号, 关注更多比赛资讯~



## 集合

### 【题目描述】

给定正整数  $n$ ，计算  $n$  个元素的集合  $\{1, 2, \dots, n\}$ ，所有非空子集的乘积取模 998 244 353 后的结果。

### 【输入格式】

一个正整数  $n$ ，代表集合大小。

### 【输出格式】

一行一个整数代表答案。

### 【样例 1 输入】

3

### 【样例 1 输出】

2160

### 【说明】

例如 3 个元素的集合有 7 个非空子集，分别为  $\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$ ，对子集内元素求和再乘积的结果就是

$$1 \times 2 \times 3 \times (1 + 2) \times (1 + 3) \times (2 + 3) \times (1 + 2 + 3) = 2160$$

### 【样例 2 输入】

11

### 【样例 2 输出】

548236960

### 【样例 3 输入】

40

### 【样例 3 输出】

133045141

### 【样例 4 输入】

150

### 【样例 4 输出】

267526432

**【备注】**

测试点编号	$n$
1 – 3	等于测试点编号
4 – 6	$\leq 20$
7 – 9	$\leq 50$
10	$\leq 200$

## 出租

### 【题目描述】

富萝莉白浅有  $n$  栋楼，编号分别为  $1 \sim n$ ，每栋楼都有  $k$  个房间以供出租，每个房间只能住一人。对于租户来说，大家都希望租到一些地理位置合适的房间。假设某个人喜好的位置是  $x$ ，那么他就只会选择在  $x \sim x + d$  这些楼中的某一个房间住下（ $d$  是本题的给定值）。

现在有  $m$  次询问，每次询问会给出两个数字  $x, y$ ，表示现在来了  $y$  个喜好位置为  $x$  的人想要租房，如果  $y$  为负数，则表示离开了  $-y$  位喜好位置为  $x$  的租户，保证离开之后喜好位置为  $x$  的租户数量不为负数。对于每次询问你都需要回答 *YES* 或 *NO* 表示目前白浅能否给每个人都分配到理想的房间。

注意：白浅可以随时更换租户的房间，但前提是新房间也要符合租户的喜好，即新房间要位于编号为  $x \sim x + d$  的楼。

大样例来喽 sample.zip

### 【输入格式】

第一行包含四个非负整数  $n, m, k, d$

接下来包含  $m$  行，每行给出两个整数  $x, y$ ，表示一次询问。

### 【输出格式】

输出  $m$  行，每行包含一个字符串 *YES* 或 *NO* 表示答案。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
YES
YES
NO
YES
```

### 【备注】

对于 1-3 的测试点，有  $n, m \leq 2000$

对于 4-5 测试点，有  $k = 1$

对于 1-10 测试点，有  $1 \leq n, m, d \leq 5 * 10^5, 0 \leq k, y \leq 10^9, 1 \leq x \leq n - d$

## 连通块

### 【题目描述】

给定一棵根为 1 的包含  $n$  个点的树，这些点分别被编号为  $1 \sim n$ 。其中第  $i$  个点的权值为  $a_i$ ， $dfs$  序为  $dfn_i$ 。

现在你需要在这棵树上找到一个连通块，连通块需要满足题目给定的  $m$  个限制，每个限制用两个正整数  $u, v$  描述。要么  $u, v$  不同时存在于这个连通块中，要么  $u, v$  在这个连通块上的  $dfs$  序不相邻。请在这棵树上找到满足题目限制条件的权值之和最大的连通块。

备注 1:

一个结点可能有多个孩子结点，这些孩子是有  $dfs$  的先后顺序的， $dfs$  顺序就是题目输入的顺序。

备注 2:

连通块的  $dfs$  序的定义：找到连通块在原树上深度最小的点开始  $dfs$ ，然后仍然根据题目读入的顺序获取每个点的  $dfs$  顺序。

大的来了（指样例）sample.zip

### 【输入格式】

第一行包含两个正整数  $n, m$ ，表示结点的个数和限制的个数；

第二行包含  $n$  个整数，第  $i$  个整数为第  $i$  个结点的权值  $val_i$  ( $-10^9 \leq val_i \leq 10^9$ )

接下来的  $n$  行里，第  $i$  行的第一个数字为一个非负整数  $s_i$ ，表示第  $i$  个结点的孩子个数，然后给出  $s_i$  个正整数按顺序表示它的所有孩子编号。

接下来的  $m$  行，每行给出两个正整数  $u, v$  ( $1 \leq u, v \leq n$ ) 描述一个约束，表示最终的连通块中  $u, v$  的  $dfs$  序不能相邻。

### 【输出格式】

输出一行一个整数表示答案。

### 【样例 1 输入】

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

5 6

2 6

**【样例 1 输出】**

12

**【备注】**

测试点编号	$n \leq$	$m \leq$	特殊性质
1 – 4	20	3	保证树随机
5 – 8	100000	5	保证树随机
9 – 12	100000	20	保证树随机，对于所有限制， 保证 $u$ 是 $v$ 在树上的第一个儿子
13 – 14	100000	20	保证树随机
15 – 16	100000	12	
17 – 20	100000	21	

## 跳棋

### 【题目描述】

白浅妹妹喜欢玩跳棋，她定义唯一合法的跳法是一个棋子  $x$  跳过一个相邻的棋子  $y$  到该直线上与  $y$  相邻的空位。她试图给一个局面能达到的所有局面计数。为了让大家都做出这道题，所以白浅妹妹给出的是一个  $1 \times n$  的棋盘。白浅妹妹会告诉你初始每个位置是否有棋子，或者她觉得这个位置无所谓有没有棋子。你需要对于每一种棋盘的可能的初始情况，求出这个局面经过若干步跳跃能达到的局面有多少种。为了减少输出量，你只需要输出每种可能初始情况对应答案之和对  $10^9 + 7$  取模的结果。

加油啊，热爱测样例的大哥哥：sample.zip

### 【输入格式】

第一行包含一个正整数  $x$ ，表示棋盘的大小为  $1 \times n$ 。第二行包含  $n$  个字符，0 表示空位，1 表示棋子，? 表示无所谓，可以是棋子也可以是空位。

### 【输出格式】

输出一行一个整数表示答案。

### 【样例 1 输入】

```
5
?0110
```

### 【样例 1 输出】

```
7
```

### 【说明】

【样例 1 解释】共有两种可能的序列。对于序列 00110，有 4 种可能，分别是 11000, 01100, 00110, 00011。对于序列 10110，有 3 种可能，分别是 11100, 10110, 10011。

大样例说明：

sample3.in 和 sample3.ans 满足 subtask 2

sample4.in 和 sample4.ans 满足 subtask 3

sample5.in 和 sample5.ans 满足 subtask 4

sample6.in 和 sample6.ans 满足 subtask 5

**【样例 2 输入】**

3

???

**【样例 2 输出】**

10

**【说明】**

三个问号本身可以产生 8 种初始局面，由初始局面 110 可以跳成 011，这是第 9 种局面；由初始局面 011 可以跳成 110，这是第 10 种局面。也就是说也许最终局面一样，但是跳出来的方式不同，则认为不同的方案。

**【备注】**

subtask1 (10pts):  $n \leq 10$

subtask2 (10pts):  $n \leq 20$ , '?' 的个数小于等于 5。

subtask3 (20pts):  $n \leq 500$ , 保证序列中全部为 '?'。

subtask4 (20pts):  $n \leq 500$ , 保证序列中不存在 '?', 且只存在一段连续的棋子。

subtask5 (40pts):  $n \leq 500$ 。对于 100% 的数据,  $n \leq 500$ 。