

# NOIP模拟赛day1

题目名称	星形	树状数组	多叉树	括号序列
题目类型	传统型	传统型	传统型	传统型
输入文件名	star.in	fenwick.in	ktree.in	bracket.in
输出文件名	star.out	fenwick.out	ktree.out	bracket.out
每个测试点时限	1.0 秒	2.0 秒	3.0 秒	3.0 秒
内存限制	512 MB	512 MB	512 MB	512 MB
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	star.cpp	fenwick.cpp	ktree.cpp	bracket.cpp
-----------	----------	-------------	-----------	-------------

编译选项

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

## 注意事项

1. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
2. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 `0`。
3. 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分隔。
4. 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。
5. 程序可使用的栈空间内存限制与题目的内存限制一致。
6. 题目不一定按照难度顺序排序，请注意掌握时间。

## 星形 (star)

### 【题目描述】

小 T 有一棵  $n$  个结点的树。他想通过删掉若干（可以为 0）个结点来使这棵树变成星形。请问有多少种不同的删点方式满足条件。

一棵树是星形当且仅当它恰好含有一个度数  $\geq 3$  的结点。

换句话说，给定一棵  $n$  个结点的树，你需要统计满足以下条件的点集  $S$  的个数。

- $S$  的导出子图中恰有一个点  $u$  满足  $\deg_u \geq 3$ ，其中  $\deg_u$  表示  $u$  的度数。

你只需要输出答案对 998244353 取模的结果。

即：统计满足条件的点集  $S$  的个数，其中点集  $S$  是联通点集

### 【输入格式】

从文件 `star.in` 中读入数据。

第一行一个整数  $n$ ，表示树的大小。

接下来  $n - 1$  行，第  $i$  行两个整数  $u_i, v_i$  表示一条树边。

### 【输出格式】

输出到文件 `star.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】

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

### 【样例 1 输出】

```
6
```

### 【样例 1 解释】

删掉的点集分别为  $\{5\}, \{6\}, \{5, 6\}, \{2\}, \{3\}, \{2, 3\}$ 。

### 【样例 2】

见选手目录下的 `star/ex_star2.in` 与 `star/ex_star2.out`。

**【样例 3】**

见选手目录下的 *star/ex\_star3.in* 与 *star/ex\_star3.out*。

**【数据范围与提示】**

对于 20% 的数据,  $n \leq 15$ 。

对于 40% 的数据,  $n \leq 2000$ 。

对于另外 10% 的数据,  $u_i = i, v_i = i + 1$ 。

对于另外 10% 的数据,  $u_i = 1, v_i = i + 1$ 。

对于另外 10% 的数据, 保证给定的树为星形。

对于 100% 的数据,  $1 \leq n \leq 5 \times 10^5$ , 保证给定的边构成一棵树。

## 树状数组 (fenwick)

### 【题目描述】

小 T 刚刚学习了树状数组，他决定练习一下。

具体来说，他想维护一个长度为  $n$  的整数序列  $[a_1, a_2, \dots, a_n]$ ，初始均为 0，执行  $m$  次操作，有如下两种操作：

1. 1 l r v: 对于每个  $i \in [l, r]$ ，将序列  $a$  中区间  $[i - f(i) + 1, i]$  里的数增加  $v$ 。
2. 2 l r: 询问序列  $a$  中区间  $[l, r]$  的和。

其中  $f(x)$  表示将  $x$  写成二进制数后，仅保留最低的 1 以及之后的 0 得到的值。例如 44 写成二进制为  $(101100)_2$ ，所以  $f(44) = (100)_2 = 4$ 。

### 【输入格式】

从文件 `fenwick.in` 中读入数据。

第一行两个整数  $n, m$ ，分别表示序列长度和操作次数。

接下来  $m$  行，每行四个整数  $1, l, r, v$  或三个整数  $2, l, r$ ，表示一次操作。

### 【输出格式】

输出到文件 `fenwick.out` 中。

对于每个询问操作，输出一行一个数表示答案。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
12
6
6
20
48
```

**【样例 1 解释】**

第一次修改完的序列为  $[3, 3, 6, 3, 3, 0, 0, 0, 0]$ 。

第二次修改完的序列为  $[7, 6, 9, 5, 6, 2, 2, 1, 1]$ 。

第三次修改完的序列为  $[8, 7, 10, 6, 7, 3, 3, 2, 2]$ 。

**【样例 2】**

见选手目录下的 *fenwick/ex\_fenwick2.in* 与 *fenwick/ex\_fenwick2.out*。

**【样例 3】**

见选手目录下的 *fenwick/ex\_fenwick3.in* 与 *fenwick/ex\_fenwick3.out*。

**【数据范围与提示】**

对于 20% 的数据,  $n, m \leq 2 \times 10^3$ 。

对于另外 10% 的数据,  $n, m \leq 10^5$ , 对于所有修改操作, 满足  $l = r$ 。

对于另外 20% 的数据,  $n, m \leq 10^5$ , 对于所有询问操作, 满足  $l = r$ 。

对于 70% 的数据,  $n, m \leq 10^5$ 。

对于 100% 的数据,  $1 \leq n, m \leq 10^6$ ,  $1 \leq l \leq r \leq n$ ,  $1 \leq v \leq 10^4$ 。

**注意：**本题的输入输出量较大，且评测环境为 windows，请使用下发的快读模板或使用自己手写的 `fread/fwrite`。不保证使用其它输入输出方式（包括普通快读）的正解能够通过本题。

## 多叉树 (ktree)

### 【题目描述】

小 T 有一个深度为  $n$  的满  $k$  叉树，树上每个结点都有一个权值。树上的叶节点从左到右编号为  $1 \sim k^n$ 。

叶节点的权值为 0 或 1，而非叶结点的权值定义为它子树内叶结点的权值总和。

开始时叶节点的权值均为 0，小 T 每次操作可以将其中一个叶节点的权值从 0 变为 1，那么经过  $k^n$  次操作后，所有叶子的权值均为 1。小 T 想知道，有多少长为  $k^n$  的操作序列满足以下条件：

- 每次操作结束后，所有非叶结点的儿子的权值的极差不超过 1。
- 有  $m$  条限制，第  $i$  条限制形如  $t_i, x_i$ ，表示第  $t_i$  次操作的叶节点必须为  $x_i$ 。

你只需要输出答案对 998244353 取模的结果。

深度为  $n$  的满  $k$  叉树是指所有非叶结点都有恰好  $k$  个儿子，并且所有叶结点的深度都为  $n$ 。特别地，根的深度为 0。

### 【输入格式】

从文件 *ktree.in* 中读入数据。

本题输入文件包含多组数据。

第一行一个整数  $T$ ，表示数据组数。

每组测试数据的第一行包含三个整数  $k, n, m$ ，分别表示树的叉数，树的深度以及限制条数。

接下来  $m$  行，第  $i$  行两个整数  $t_i, x_i$ ，表示一条限制。

### 【输出格式】

输出到文件 *ktree.out* 中。

对每组测试数据输出一行一个整数，表示答案对 998244353 取模的结果。

### 【样例 1 输入】

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

2 6  
8 3

### 【样例 1 输出】

4  
0  
2  
96

### 【样例 1 解释】

第一组数据中合法的操作序列有  $[2, 3, 1, 4], [2, 3, 4, 1], [2, 4, 1, 3], [2, 4, 3, 1]$ 。其中操作序列里的第  $i$  个数表示第  $i$  次操作的叶子的编号。

第二组数据中没有合法的操作序列。

第三组数据中合法的操作序列有  $[2, 3, 1, 4, 5], [5, 3, 1, 4, 2]$ 。

### 【样例 2】

见选手目录下的 *ktree/ex\_ktree2.in* 与 *ktree/ex\_ktree2.out*。

### 【样例 3】

见选手目录下的 *ktree/ex\_ktree3.in* 与 *ktree/ex\_ktree3.out*。

### 【数据范围与提示】

对于 20% 的数据,  $T \leq 20, k^n \leq 9$ 。

对于另外 10% 的数据,  $m \leq 1$ 。

对于另外 20% 的数据,  $n = 2$ 。

对于另外 20% 的数据,  $k = 2$ 。

对于 100% 的数据,  $T \leq 2 \times 10^5, 2 \leq k \leq 10^6, n \geq 1, 0 \leq m \leq 2 \times 10^5, \sum m \leq 2 \times 10^5, 1 \leq t_i, x_i \leq k^n \leq 10^{18}$ , 对于每组数据保证  $t_i$  互不相同,  $x_i$  互不相同。

## 括号序列 (bracket)

### 【题目描述】

小 T 有一个长为  $2n$  的括号序列，用一个长为  $2n$  的序列  $a$  表示。序列中有  $n$  种不同的括号，编号为  $1 \sim n$ 。每种括号在序列中恰好出现了两次，第一次出现为左括号，第二次出现为右括号。例如序列  $1\ 2\ 3\ 2\ 3\ 1$  就代表了括号序列  $[{\{()}}]$ 。

“符合规范的超级括号序列”的定义如下：

1. 空序列， $S\ S$  均是符合规范的超级括号序列，其中  $S$  表示一个在  $[1, n]$  之间的正整数（以下规则中的  $S$  均为此含义）；
2. 如果  $A$  为非空的符合规范的超级括号序列，那么  $S\ A\ S$  为符合规范的超级括号序列；
3. 如果  $A$  和  $B$  均为符合规范的超级括号序列，那么  $A\ B$  为符合规范的超级括号序列；
4. 所有符合规范的超级括号序列均可通过上述 3 条规则得到。

小 T 认为符合规范的超级括号序列非常炫酷，可惜他目前得到的这个序列不一定满足条件。于是他决定将这个序列划分成两个子序列，使得每个子序列都是符合规范的超级括号序列。

可惜小 T 对如何划分毫无头绪，于是只好请求你来帮忙。你需要给出一种满足条件的划分方案，或输出无解。

### 【输入格式】

从文件 `bracket.in` 中读入数据。

本题输入文件包含多组数据。

输入的第一行包含一个整数  $T$ ，表示测试数据的组数。

每组测试数据的第一行包含一个正整数  $n$ ，第二行包含  $2n$  个用空格隔开的整数  $a_1, a_2, \dots, a_{2n}$ 。

### 【输出格式】

输出到文件 `bracket.out` 中。

对每个测试数据输出一行答案。

如果无解，输出一行  $-1$ ，否则输出一行一个长度为  $2n$  的、由字符  $1$  或  $2$  构成的字符串（不含空格），其中  $1$  表示该元素位于第一个子序列， $2$  表示该元素位于第二个子序列。

本题使用自定义校验器检验你的答案是否正确，因此若有多种满足条件的方案，你只需要输出任意一种。

### 【样例 1 输入】

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



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

### 【样例 1 输出】

```
212122
-1
111111
2212121111
```

### 【样例 1 解释】

第一组数据中将序列 1 2 3 2 3 1 划分为 2 2 和 1 3 3 1。

第二组数据不存在合法的划分方式。

第三组数据中将序列 1 2 3 3 2 1 划分为 1 2 3 3 2 1 和空序列。注意某个子序列可以为空。

第四组数据中将序列 5 2 4 2 1 5 1 4 3 3 划分为 4 1 1 4 3 3 和 5 2 2 5。

### 【样例 2】

见选手目录下的 *bracket/ex\_bracket2.in* 与 *bracket/ex\_bracket2.out*。

### 【样例 3】

见选手目录下的 *bracket/ex\_bracket3.in* 与 *bracket/ex\_bracket3.out*。

### 【数据范围与提示】

对于 20% 的数据,  $T = 1, n \leq 20$ 。

对于 40% 的数据,  $\sum n \leq 2000$ 。

对于 60% 的数据,  $\sum n \leq 10^5$ 。

对于另外 20% 的数据, 保证  $a_1, a_2, \dots, a_n$  为一个  $1 \sim n$  的排列。

对于 100% 的数据,  $1 \leq T \leq 2 \times 10^4, 1 \leq n \leq 10^6, \sum n \leq 1.5 \times 10^6, 1 \leq a_i \leq n$ , 保证序列  $a$  中  $1 \sim n$  每个数恰好出现了两次。

**注意：**本题的输入输出量较大, 建议使用较快的输入输出方式。