

# NOIP模拟赛day3

|         |          |                 |            |          |
|---------|----------|-----------------|------------|----------|
| 题目名称    | 插旗       | 括号序列加强版         | 研表究明       | 硬币       |
| 题目类型    | 传统型      | 传统型             | 传统型        | 传统型      |
| 输入文件名   | flag.in  | bracketplus.in  | string.in  | coin.in  |
| 输出文件名   | flag.out | bracketplus.out | string.out | coin.out |
| 每个测试点时限 | 1.0 秒    | 1.0 秒           | 1.0 秒      | 2.0 秒    |
| 内存限制    | 512 MB   | 512 MB          | 512 MB     | 512 MB   |
| 测试点数目   | 10       | 10              | 20         | 4        |
| 测试点是否等分 | 是        | 是               | 是          | 否        |

## 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。

## 插旗 (flag)

### 【题目描述】

小 T 很喜欢立 flag，所以他决定要在每个地方都立上 flag。

具体来说，现在有  $n$  个地点，用  $1 \sim n$  标号，有  $n - 1$  条无向道路将这些地点连通，也就是说它们构成了一棵树。两个地点之间的距离定义为它们最短路径经过的道路数。

小 T 最开始在 1 号地点。每次，小 T 会找到所有距离他不超过  $k$  且没有被插上旗的地点，从中选择一个离他最近的地点（如果有多个距离他最近的地点他可以任意选择其中一个），然后走到那个地点插上旗子。小 T 需要将这  $n$  个地点都插上旗子，且他最后一个插旗的位置到 1 号地点的距离不能超过  $k$ 。

现在小 T 还没有确定参数  $k$  的取值，请你求出最小的正整数  $k$ ，使得存在一个满足上述条件的插旗方案。

### 【输入格式】

从文件 *flag.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。

对于每组数据，第一行一个整数  $n$  表示地点个数。

接下来  $n - 1$  行，第  $i$  行两个整数  $u_i, v_i$  表示一条连接  $u_i$  号地点与  $v_i$  号地点的无向道路。

### 【输出格式】

输出到文件 *flag.out* 中。

对每个测试数据输出一行答案。

对于每组数据输出一行一个正整数  $k$  表示答案。

### 【样例 1 输入】

```
3
3
1 2
1 3
4
1 2
2 3
3 4
8
1 2
2 3
3 4
```

1 5  
5 6  
6 7  
5 8

### 【样例 1 输出】

2  
3  
3

### 【样例 1 解释】

第一组数据中  $k = 2$  时小 T 的走法如下：

1. 小 T 在 1 号地点，距离他不超过  $k$  且没有插过旗的地点中，离他最近的显然是 1 号地点，于是他会在 1 号地点插上旗；
2. 小 T 在 1 号地点，可行的地点有 2,3 号地点，小 T 走向 2 号地点并插上旗；
3. 小 T 在 2 号地点，可行的地点有 3 号地点，小 T 走向它并插上旗。
4. 小 T 在 3 号地点，这时他离 1 号地点的距离不超过  $k$ ，满足条件。

第二组数据中，小 T 的走法为  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1$ 。

第三组数据中，小 T 的走法为  $1 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1$ 。

### 【样例 2】

见选手目录下的 *flag/ex\_flag2.in* 与 *flag/ex\_flag2.out*。

### 【样例 3】

见选手目录下的 *flag/ex\_flag3.in* 与 *flag/ex\_flag3.out*。

### 【数据范围与提示】

对于 20% 的数据， $T = 1$ ， $n \leq 10$ 。

对于另外 20% 的数据，每个地点最多被两条道路所连接。

对于另外 20% 的数据， $u_i = \lfloor \frac{i+1}{2} \rfloor$ ， $v_i = i + 1$ 。

对于 100% 的数据， $1 \leq T \leq 10^4$ ， $2 \leq n \leq 2 \times 10^5$ ， $\sum n \leq 2 \times 10^5$ ，保证道路形成一棵树。

## 括号序列加强版 (bracketplus)

### 【题目描述】

小 T 得到了一个序列  $a$ 。如果一个序列能被看作一个括号序列，那么小 T 称这样的序列为符合规范的超级括号序列。

“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 想知道有多少  $a$  的非空子区间使得它是符合规范的超级括号序列。

### 【输入格式】

从文件 *bracketplus.in* 中读入数据。

第一行一个整数  $n$  表示序列长度。

第二行  $n$  个整数  $a_i$  表示序列  $a$ 。

### 【输出格式】

输出到文件 *bracketplus.out* 中。

仅一行一个整数表示答案。

### 【样例 1 输入】

```
9
3 1 2 2 1 6 6 3 3
```

### 【样例 1 输出】

```
8
```

### 【样例 1 解释】

这 8 个子区间分别是  $[1, 8], [2, 5], [2, 7], [2, 9], [3, 4], [6, 7], [6, 9], [8, 9]$ 。

### 【样例 2】

见选手目录下的 *bracketplus/ex\_bracketplus2.in* 与 *bracketplus/ex\_bracketplus2.out*。

**【样例 3】**

见选手目录下的 *bracketplus/ex\_bracketplus3.in* 与 *bracketplus/ex\_bracketplus3.out*。

**【数据范围与提示】**

对于 10% 的数据,  $n \leq 300$ 。

对于 30% 的数据,  $n \leq 5000$ 。

对于另外 20% 的数据, 保证每种数字最多在  $a$  中出现两次, 且  $a$  本身是符合规范的超级括号序列。

对于另外 20% 的数据, 每个  $a_i$  在  $1, 2, 3, 4$  中独立均匀随机。

对于 100% 的数据,  $1 \leq n \leq 3 \times 10^5$ ,  $1 \leq a_i \leq n$ 。

## 研表究明 (string)

### 【题目背景】

研表究明,汉字的序顺并不定一影阅响读。

### 【题目描述】

小 T 觉得打乱句子中字母的顺序很有意思。为了简化问题,我们只考虑 01 串。

定义长为  $n$  的 01 串  $b$  能被长为  $n$  的 01 串  $a$  生成当且仅当存在一个  $1 \sim n$  的排列  $p_i$ , 满足对于每个  $1 \leq i \leq n$ , 有  $-1 \leq p_i - i \leq k$  且  $b_{p_i} = a_i$ 。其中  $k$  为给定的正整数。

定义一个 01 串的美观度为它能生成不同的 01 串的数量。

给定一个 01 串  $A$ , 你要求出它的所有非空子区间的美观度之和。答案对 998244353 取模。

由于串  $A$  太长了, 我们会给定一个 01 串  $S$  以及正整数  $t$ , 串  $A$  恰好为  $t$  个  $S$  依次拼接而成。

### 【输入格式】

从文件 `string.in` 中读入数据。

输入的第一行包含两个整数  $k, t$ , 含义见题目描述。

第二行一个 01 串  $S$ 。串  $A$  可由  $t$  个  $S$  拼接得到。

### 【输出格式】

输出到文件 `string.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】

```
2 1
001
```

### 【样例 1 输出】

```
8
```

### 【样例 1 解释】

对于区间  $[1, 1]$ , 它能生成的 01 串有 0。

对于区间  $[2, 2]$ , 它能生成的 01 串有 0。

对于区间  $[3, 3]$ , 它能生成的 01 串有 1。

对于区间  $[1, 2]$ , 它能生成的 01 串有 00。

对于区间  $[2, 3]$ , 它能生成的 01 串有 01, 10。

对于区间  $[1, 3]$ , 它能生成的 01 串有 001, 010。

**【样例 2】**

见选手目录下的 *string/ex\_string2.in* 与 *string/ex\_string2.out*。

**【样例 3】**

见选手目录下的 *string/ex\_string3.in* 与 *string/ex\_string3.out*。

**【数据范围与提示】**

对于 20% 的数据,  $t = 1$ ,  $|S| \leq 15$ 。

对于 40% 的数据,  $t = 1$ ,  $|S| \leq 5000$ 。

对于 60% 的数据,  $t = 1$ 。

对于另外 20% 的数据,  $k = 1$ 。

对于 100% 的数据,  $1 \leq |S| \leq 2 \times 10^6$ ,  $1 \leq t \leq 10^9$ ,  $1 \leq k \leq |S| - 1$ 。

## 硬币 (coin)

### 【题目描述】

初始有两个空的队列， $2n$  个人站成一排，每个人以上帝之手扔一个质地均匀的硬币（抛出正反的概率相同）。然后按照编号  $i = 1, 2, \dots, 2n$  依次考虑每个人：

- 如果第  $i$  个人的硬币是正面：若第一个队列的人数  $< n$ ，则第  $i$  个人到第一个队列；否则到第二个队列。
- 如果第  $i$  个人的硬币是反面：若第二个队列的人数  $< n$ ，则第  $i$  个人到第二个队列；否则到第一个队列。

显然最终队列各有  $n$  人。现在有  $m$  次询问，每次询问给定一个  $\{1, 2, \dots, 2n\}$  的子集  $S$ ，询问编号在  $S$  内的人全都在一个队列的概率是多少。答案对 998244353 取模。

### 【输入格式】

从文件 `coin.in` 中读入数据。

第一行两个整数  $n, m$ 。

接下来  $m$  行，每行第一个整数  $k$  表示询问的  $S$  的大小，接下来紧跟  $k$  个数  $1 \leq a_1 < a_2 < \dots < a_k \leq 2n$ ，表示集合  $S$  中的元素。

### 【输出格式】

输出到文件 `coin.out` 中。

共  $m$  行，每行一个整数表示询问的答案对 998244353 取模后的值。

### 【样例 1 输入】

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



**【样例 1 输出】**

```
499122177
748683265
748683265
748683265
748683265
499122177
```

**【样例 2 输入】**

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

**【样例 2 输出】**

```
935854081
623902721
374341633
935854081
686292993
```

**【样例 3】**

见选手目录下的 *coin/coin3.in* 与 *coin/coin3.ans*。

**【数据范围】**

对于 100% 的数据,  $2 \leq k \leq n \leq 10^5, 1 \leq m \leq 10^5, \sum k \leq 2 \times 10^5$ 。

| 子任务编号 | $n \leq$ | 特殊性质    | 子任务分值 |
|-------|----------|---------|-------|
| 1     | 20       | 无       | 20    |
| 2     | 2000     | $m = 1$ | 20    |
| 3     | $10^5$   | $k = 2$ | 20    |
| 4     | $10^5$   | 无       | 40    |