

# 2024 年广东省重点中学信息学邀请赛 (GDKOI 2024)

普及组 第二试

2024 年 1 月 7 日

## 注意事项

- 1. 严格按照题目所要求的格式进行输入、输出，否则严重影响得分。
- 2. 题目测试数据有严格的时间限制，超时不得分。
- 3. C/C++ 中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 `0`。
- 4. 输入文件格式不用判错；输入输出文件名均已给定，不用键盘输入。
- 5. 评测环境为 NOI 系列活动标准竞赛环境，编译器版本为 `g++ 9.4.0`。
- 6. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
- 7. 对于 C++ 选手，64 位整数输入输出格式为 `%lld`。
- 8. 选手提交的程序源文件必须不大于 100KB。
- 9. 对于 C++ 语言的编译选项为 `-O2 -std=c++14`

| 试题名称  | 捉迷藏      | 读书       | 正方形扩展     | EX 中缀表达式  |
|-------|----------|----------|-----------|-----------|
| 提交文件名 | game.cpp | book.cpp | sqrex.cpp | exmid.cpp |
| 输入文件名 | game.in  | book.in  | sqrex.in  | exmid.in  |
| 输出文件名 | game.out | book.out | sqrex.out | exmid.out |
| 时间限制  | 3 秒      | 1 秒      | 1 秒       | 2 秒       |
| 空间限制  | 512 MB   | 512 MB   | 512 MB    | 512 MB    |
| 满分    | 100      | 100      | 100       | 100       |

## 第一题 捉迷藏

提交文件: `game.cpp`  
输入文件: `game.in`  
输出文件: `game.out`  
时间空间限制: 3 秒, 512 MB

Zayin 和 Ziyin 正在玩有趣的捉迷藏游戏。

该游戏在一颗具有  $n$  个节点（编号从 1 到  $n$ ）的树上进行。

在游戏的开始, Zayin 在节点  $a$ , 而 Ziyin 在节点  $b$ 。他们轮流操作, Zayin 先移动。在每次移动中, Zayin 能移动到距离当前所在点不超过  $da$  的节点上, 而 Ziyin 能移动到距离当前所在点不超过  $db$  的节点上（注意可以保持在当前点不动）。

当某次移动后, 其中一人抓住了另外一人, 即移动到了另外一人的节点上, 则游戏结束, 被抓住的人输掉游戏。

当 Zayin 和 Ziyin 都按最优策略移动的话, 谁会是最后赢家呢。

注解:

- 一颗具有  $n$  个节点的树是指一个具有  $n$  个节点,  $n - 1$  条边的连通无向图。
- 树上两个节点的距离定义为连接该两点的最短路径所包含的边数。

## 输入格式

每个测试点包含多个测试用例。

第一行包含两个整数  $d, t$ , 表示测试点编号, 和测试用例的数量。每个测试用例的描述如下。

第一行包含两个整数  $n, q$  — 分别为顶点数、询问数。

接下来  $n-1$  行每行包含两个整数  $u, v$  ( $1 \leq u, v \leq n, u \neq v$ ), 表示顶点  $u$  和  $v$  之间具有一条直接相连的边, 保证这些边形成一棵树。

接下来  $q$  行每行包含四个整数  $a, b, da, db$  ( $1 \leq a, b, da, db \leq n$ ) 作为一次游戏, 分别表示 Zayin 初始节点、Ziyin 初始节点、Zayin 最大移动距离、Ziyin 最大移动距离。

## 输出格式

对于每个测试用例的每次游戏, 输出一行 "Zayin" 或 "Ziyin" 表示最后赢家, 特别地如果在  $10^{10^5}$  轮内游戏没有仍结束, 则输出 "Draw" 表示平局。

样例数据

| game.in | game.out |
|---------|----------|
| 1 2     | Ziyin    |
| 6 5     | Zayin    |
| 2 3     | Zayin    |
| 2 6     | Ziyin    |
| 2 1     | Ziyin    |
| 4 3     | Zayin    |
| 5 1     | Zayin    |
| 5 4 1 2 | Zayin    |
| 6 4 4 3 | Draw     |
| 1 4 5 4 | Ziyin    |
| 5 2 1 4 |          |
| 2 5 1 5 |          |
| 4 5     |          |
| 1 4     |          |
| 3 4     |          |
| 2 4     |          |
| 4 2 2 3 |          |
| 4 3 2 2 |          |
| 4 3 3 3 |          |
| 1 2 1 1 |          |
| 1 2 1 2 |          |

数据范围

保证所有测试用例的  $n$  之和不超过  $10^6$ ， $q$  之和不超过  $10^6$ 。

| 数据点编号    | $\sum n \leq$ | 特殊性质 |
|----------|---------------|------|
| 1        | 10            | 无    |
| 2        | 100           | q=1  |
| 3        | 100           | 无    |
| 4        | $10^4$        | q=1  |
| 5        | $10^4$        | 无    |
| 6        | $10^6$        | q=1  |
| 7,8,9,10 | $10^6$        | 无    |

## 第二题 读书

提交文件: book.cpp  
输入文件: book.in  
输出文件: book.out  
时间空间限制: 1 秒, 512 MB

Zayin 是一个热爱读书的学生。

最近, Zayin 收到了一本有  $n$  个章节的书, 其中每个章节  $i$  都有一个限制: 她必须至少阅读了其他  $a_i$  个章节, 才能够获取足够的智慧来读懂该章节。

每天, Zayin 都会从头到尾开始阅读这本书。对于她还不能读懂的章节 (由于限制) 或是已经阅读过的章节, Zayin 会在那天跳过它们。

现在, Zayin 想要知道至少需要多少天才能阅读完所有的  $n$  个章节。

### 输入格式

第一行包含两个整数  $d, n$ , 表示测试点编号和章节数。

第二行包含  $n$  个整数  $a_i$  ( $0 \leq a_i < n$ ), 表示限制。

### 输出格式

输出一行包含一个整数, 表示最少需要的天数。

如果 Zayin 无法阅读完所有的  $n$  个章节, 输出  $-1$ 。

### 样例数据

| book.in                     | book.out |
|-----------------------------|----------|
| 1 10<br>3 4 0 6 1 1 0 8 6 3 | 2        |

### 数据范围

本题使用子任务捆绑测试。

对于所有测试数据, 保证  $1 \leq n \leq 5 \times 10^5, 0 \leq a_i < n$ 。

Subtask 1(10%) :  $1 \leq n \leq 10, d = 1$ 。

Subtask 2(10%) :  $1 \leq n \leq 500, d = 2$ 。

Subtask 3(20%) :  $1 \leq n \leq 5000, 3 \leq d \leq 4$ 。

Subtask 4(20%) :  $1 \leq n \leq 10^5, 5 \leq d \leq 6$ 。

Subtask 5(40%) :  $1 \leq n \leq 5 \times 10^5, 7 \leq d \leq 10$ 。

## 第三题 正方形扩展

提交文件: sqrex.cpp  
输入文件: sqrex.in  
输出文件: sqrex.out  
时间空间限制: 1 秒, 512 MB

现在, 在笛卡尔坐标系 (无限大二维平面) 上有  $n$  个种类互不相同的细菌, 它们所在的坐标也互不相同。随着时间的增加, 细菌们不断繁殖, 以正方形的形状、用相同的正方形扩张速度, 同时扩张自己的领地。具体来说对于任意时刻  $t$ 、平面上任意一点  $p$ , 假设该点  $p$  上存在第  $i$  种细菌, 那么有以下两种情况:

- 如果以点  $p$  为中心的任意正方形都含有其他种类的细菌, 则该点的细菌将不会扩张 (可以称之为“接触抑制”)。
- 如果存在一个以  $p$  为中心的正方形不含有其他种类的细菌, 则该点的细菌将会进行扩张。

注意, 扩展出去的同种细菌也具备一样的扩展能力。

以下是一些简单的关于正方形扩展的例子:

若初始时, 平面只有唯一的一个细菌位于  $(0, 0)$ , 那么过一个单位时间后, 这一类细菌将占领  $(1, 1)$   $(1, -1)$   $(-1, -1)$   $(-1, 1)$  围成的正方形。

若初始时, 平面有两个细菌分别位于  $(0, 0)$  和  $(1, 0)$ , 那么最终  $(0.5, 0)$  会成为他们领地的分界线, 一开始位于  $(0, 0)$  的细菌会占领  $(0.5, 0)$  左侧的全部区域, 位于  $(1, 0)$  的细菌会占领  $(0.5, 0)$  右侧的全部区域。

现在询问对于第  $i$  种细菌, 询问其占领面积能否趋于无穷大。

### 输入格式

第一行一个正整数  $n$  ( $1 \leq n \leq 10^6$ ) 表示细菌母体的数量。

接下来输入  $n$  行, 每行输入两个整数, 表示点的坐标  $(x_i, y_i)$ , 即种类为  $i$  的细菌母体的位置。

### 输出格式

输出一个长度为  $n$  的 01 串, 对于其中第  $i$  个数字, 1 表示种类为  $i$  的细菌的占领面积可以扩张到无穷大, 0 则表示最终面积有限。

## 样例数据

| sqrex.in                                                     | sqrex.out |
|--------------------------------------------------------------|-----------|
| 5<br>0 0<br>2 0<br>2 2<br>0 2<br>1 1                         | 11110     |
| 3<br>-2 0<br>0 0<br>2 0                                      | 111       |
| 7<br>-7 -8<br>5 -9<br>1 -5<br>9 -4<br>-8 3<br>-2 -3<br>-4 -6 | 1101110   |

## 样例解释

在第二个样例，点  $(0,0)$  最终拥有的领地是直线  $x = -1$  与  $x = 1$  夹的中间部分，面积趋于无穷大。

## 数据范围

对于 25% 数据,  $n \leq 10^2$ 。

对于 50% 数据,  $n \leq 10^3$ 。

对于 75% 数据,  $n \leq 10^5$ 。

对于 100% 数据,  $n \leq 10^6, -10^9 \leq x_i, y_i \leq 10^9$ 。

## 第四题 EX 中缀表达式

提交文件: `exmid.cpp`  
 输入文件: `exmid.in`  
 输出文件: `exmid.out`  
 时间空间限制: 2 秒, 512 MB

Alice 最近学习了表达式, 现在他想写一个属于自己的表达式计算器。

规定:

- 单个数字: 由若干数字 (至少一个) 连续拼接, 最后必须跟一个 `'` 字符 (不含引号) 例如: `00123. 0. 789.` 都是合法的数字
- 单个操作符: 由若干数字连续拼接, 最后跟一个字符
- 操作符的字符: 表示该操作符的操作。该字符必须是 `+`、`*`、`^` 之一, 分别表示加法, 乘法和乘方。特别约定  $0^0 = 1$ 。

例如: `000000789+,123^`, 如果前面的数值合法, 那么它们就都是合法的操作符

- 操作符的数字: 表示该操作符的优先级, 优先级的取值是  $[1, n]$  之间的正整数, 数字越大表示优先级越高。
- 对于优先级相同的操作符, 题目将给出一个长度为  $n$  的 01 串  $C$ , 用于说明对应优先级的操作符之间, 是左结合还是右结合。

其中 0 表示左结合, 1 表示右结合。

例如  $C = "111011"$ , 其中第 4 个字符为 `'0'`, 表示优先级为 4 的操作符是左结合的。

- 左结合: 表示该运算符从左往右计算。下面给出左结合的例子:  $1.4 + 2.4^3.4^4$ . 等价于  $((1.4 + 2.)4^3.)4^4$ , 其结果与  $((1 + 2)^3)^4$  相同
- 右结合: 表示该运算符从右往左运算。下面给出右结合的例子:  $1.6 + 2.6^3.6^4$ . 等价于  $1.6 + (2.6^{(3.6^4.)})$ , 其结果与  $1 + (2^{(3^4)})$  相同。
- 中缀表达式:

1. 单个数字是合法的中缀表达式
2. 若  $A$  是合法的中缀表达式, 则  $(A)$  也是合法的中缀表达式
3. 若  $A$ 、 $B$  均是合法的中缀表达式,  $c$  是合法的单个操作符, 则  $A c B$  也是合法的中缀表达式
4. 其余情况均不合法

现在给出一个长度为  $n$  的 01 串  $C$ , 用于说明, 相同优先级的操作符之间, 是左结合还是右结合。

给出一个中缀表达式, 判断该表达式是否合法, 不合法则输出 `"error"` (不包括引号), 合法则输出该表达式的值对 998244353 取模的结果。

### 输入格式

第一行一个正整数  $n$ 。

第二行一个长度为  $n$  的 01 串  $C$ 。

第三行一个字符串  $S$ , 表示一个中缀表达式。

保证表达式中不会存在空格。

输出格式

若中缀表达式不合法则输出”error”，否则输出该表达式的值对 998244353 取模的结果。

样例数据

| exmid.in                                | exmid.out |
|-----------------------------------------|-----------|
| 2<br>01<br>1.2+2.1^3.2*4.2^(5.2*6.)2+7. | 243640717 |

样例解释

$$243640717 = ((1 + 2)^{(3 * (4^{((5 * 6) + 7))})}) \bmod 998244353$$

数据范围

- 特殊性质 1：不会出现 ^ 字符。
- 特殊性质 2：不会出现 + 和 \* 字符。
- 特殊性质 3：不会出现 ( 和 )。
- 对于 8% 的数据， $n = 1, 1 \leq |S| \leq 100$ , 且满足特殊性质 1 和特殊性质 3。
- 对于另外 8% 的数据， $n = 1, 1 \leq |S| \leq 100$ , 且满足特殊性质 1。
- 对于另外 20% 的数据， $n = 1, 1 \leq |S| \leq 100$ , 且满足特殊性质 2 和特殊性质 3。
- 对于另外 24% 的数据， $n = 1, 1 \leq |S| \leq 1000$ , 且满足特殊性质 2。
- 对于 100% 的数据， $1 \leq n \leq 9, 1 \leq |S| \leq 10000$ 。