

# 提高组模拟赛

## 一、题目概览

中文题目名称	湮灭反应	树的计数	我没有说谎	美食家
英文题目名称	min	tree	lie	foodie
可执行文件名	min	tree	lie	foodie
输入文件名	min.in	tree.in	lie.in	foodie.in
输出文件名	min.out	tree.out	lie.out	foodie.out
时间限制	1s	1s	1s	1s
空间限制	512MB	512MB	512MB	512MB
测试点数目	10	10	10	10
测试点分值	10	10	10	10
题目类型	传统	传统	传统	传统
比较方式	全文比较	全文比较	全文比较	全文比较
是否有部分分	是	是	是	是

## 二、注意事项：

- 1.文件名（程序名和输入输出文件名）必须使用小写。
- 2.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。
- 3.选手不得使用 SSH 等命令。
- 4.选手不得使用内嵌汇编，#pragma等指令。
- 5.评测时使用 C++14 环境，同时开启 O2 优化，栈空间和内存限制相同。

# T1 湮灭反应

## 题目描述

物质与其所对应的反物质碰撞后消失并产生高能光子（ $\gamma$  射线）等能量的过程，例如质子与反质子，电子与反电子的碰撞，称之为湮灭反应。在基本情况下，湮灭就是完全的质能转换过程，湮灭的两个粒子会释放全部的能量同归于尽。

现在小明乘坐宇宙飞船遨游太空，发现了太空中有  $n$  团排成一排的正反物质，小明将这些团从左到右编号为  $1, 2, \dots, n$ ，其中第  $i$  团物质含有  $|a_i|$  个物质，如果  $a_i$  为正，则表示该团含有  $a_i$  个正物质，如果  $a_i$  为负，则表示该团含有  $|a_i|$  个反物质。

小明可以操控这些物质，他想要选择一段编号连续的物质团（不能什么都不选），然后让这些正反物质进行湮灭反应。注意，正反物质反应会两两抵消。例如，3 个正物质和 4 个反物质反应，还会剩下 1 个反物质。

小明觉得太空的物质太多了，想要使得选择出来的物质团进行反应后，剩余的物质数量**最小**。同时，小明为了展示自己的能力，想要在满足上述剩余物质数量最小的情况下，选择**尽可能长**的物质团出来。

## 输入格式

第一行，输入一个整数  $n$  表示太空中物质团的数量。

接下来有  $n$  行，第  $i + 1$  行，输入一个整数  $a_i$  描述编号为  $i$  的物质团。

## 输出格式

第一行输出一个整数，小明操作后剩余的物质数量的最小值。

第二行输出一个整数，表示在满足物质数量最小的情况下，能够选择的最长的物质团的长度。

## 样例 #1

### 样例输入 #1

```
6
70
-30
-20
60
80
-145
```

### 样例输出 #1

```
5
3
```

## 样例 #2

### 样例输入 #2

```
4
-10 10 -10 10
```

### 样例输出 #2

```
0
4
```

## 提示

【样例 1 解释】

物质团分别为 70, -30, -20, 60, 80, -145, 其中选择编号  $[2, 4]$  的物质团, 正负物质产生反应后, 会剩余 10 个正物质; 选择编号  $[4, 6]$  的物质团产生反应后会剩余 5 个负物质。可以发现, 这种情况是使得最后剩余物质数量最少的一种选择, 该选择的物质团区间长度为 3。

【数据范围】

对于 40% 的数据,  $n \leq 4000$ 。

对于 100% 的数据,  $n \leq 10^5$ , 且保证任意  $|a_i| \leq 10^{10}$ 。

值得注意的是, 数据中, 有大部分数据在满足剩余物质数量最少的前提下, 得到的最长物质团长度是唯一的。

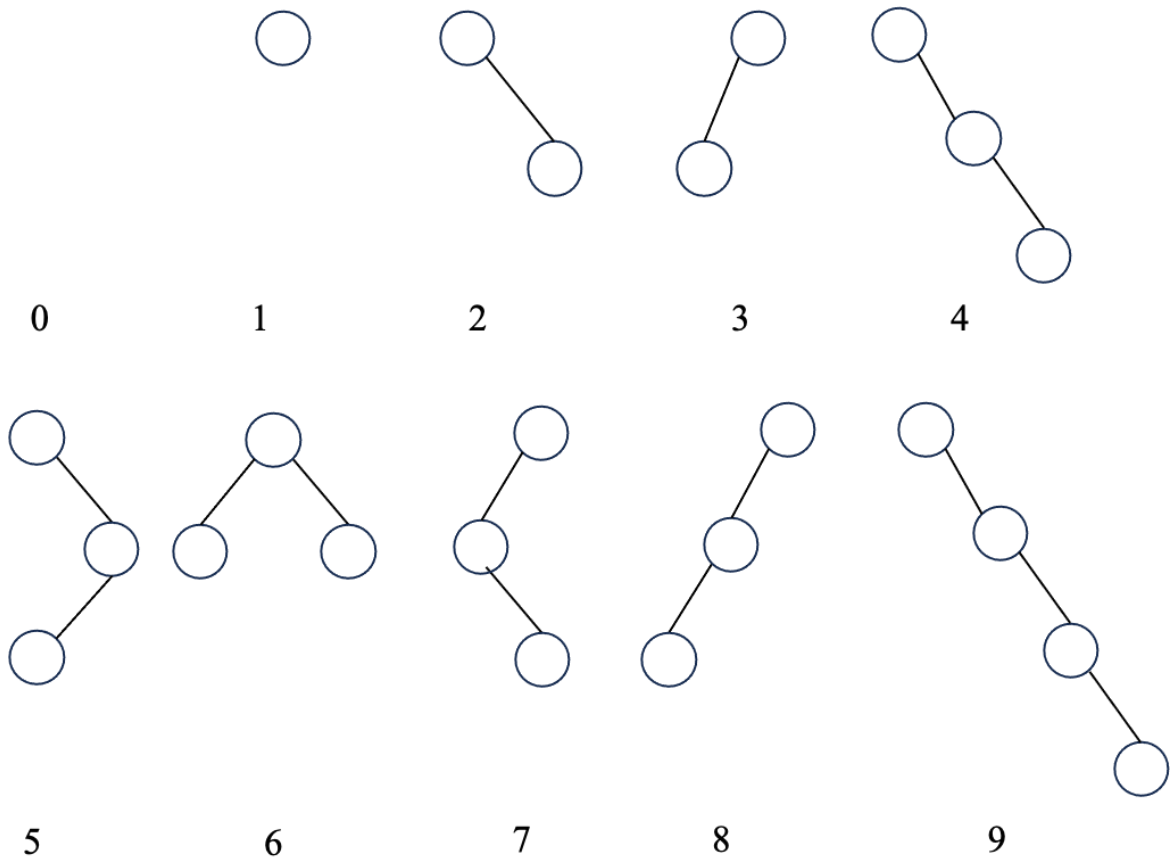
## T2 树的计数

### 题目描述

我们都知道二叉树有多种形态, 对于一系列的二叉树, 我们用下列规则对其进行编号:

1. 空二叉树编号为 0;
2. 仅含一个结点的二叉树编号为 1;
3. 结点数为  $k$  的二叉树的编号小于结点数为  $k + 1$  的二叉树的编号;
4. 对于结点数量相同的二叉树  $T_1, T_2$ , 如果  $T_1$  的左子树编号大于  $T_2$  的左子树编号或者  $T_1, T_2$  的左子树编号相同但  $T_1$  的右子树编号大于  $T_2$  的右子树编号, 那么二叉树  $T_1$  的编号大于二叉树  $T_2$  的编号。

例如, 按照上述规则进行编号的前 10 棵二叉树如下图所示:



- 其中编号为 3 的二叉树左子树是仅含一个结点的二叉树, 编号为 2 的二叉树左子树为空, 满足条件 4 中左子树编号大的整棵树编号就大。

现在给你一个正整数  $n$ ，表示二叉树的编号，让你画出编号为  $n$  的二叉树的形状。

## 输入格式

输入仅包含一行一个正整数  $n$ ，表示二叉树的编号。

## 输出格式

输出仅一行，表示编号为  $n$  的二叉树的形状。

二叉树形状用下列字符串表示：

1. 如果是仅包含一个结点的二叉树，则直接输出 `x`
2. 如果二叉树的左、右子树编号分别为  $L$  和  $R$ ，已知  $L$  和  $R$  的输出形式分别为  $L'$  和  $R'$ ，那么输出 `(L')x(R')`，让左子树为空时，输出 `x(R')`，当右子树为空时，输出 `(L')x`。

## 样例 #1

### 样例输入 #1

```
5
```

### 样例输出 #1

```
x((x)x)
```

## 样例 #2

### 样例输入 #2

```
20
```

### 样例输出 #2

```
((x)x(x))x
```

## 样例 #3

### 样例输入 #3

```
237074288
```

### 样例输出 #3

```
x((((x)x)x(x))x((((x)x((x(x))x)x))x)x(x))x))
```

## 提示

【样例 1 解释】

编号为 5 的二叉树见题目图，根结点只有右子树，所以为  $X(R')$ ，右子树的根节点只有左子树，所以  $R'$  为  $(X)X$ ，整棵树为  $X((X)X)$ 。

【数据范围】

对于 30% 的数据，满足  $1 \leq n \leq 5000$ ;

对于 100% 的数据，满足  $1 \leq n \leq 5 \times 10^8$ 。

## T3 我没有说谎

### 题目描述

小明参加了一场大型的 "欺诈游戏"，现在已经来到了最后一轮环节，最后一个环节还剩下  $n$  个人，编号为  $1, 2, \dots, n$ 。小明只要胜出，就能获得终极大奖 1,000 万。

本轮游戏开始前，主办方会在大屏幕放映随机生成的  $n$  个人的分数，也就是说大家都知道彼此的分数。在看完所有人的分数后，主办方要求每一个参与游戏的人，都说一句有几个人分数比我高，有几个人分数比我低，当然，这句话可以不是真实的，可以说谎。

每个人说完后，主办方收集了每一个人的回答，具体地，编号为  $i$  的人说的是，"有  $a_i$  个人分数比我高，有  $b_i$  个人分数比我低"。

现在问， $n$  个人中最少有几个人在说谎，如果小明回答对了这个问题，就可以获得大奖，请你帮帮小明。

### 输入格式

输入第一行一个整数，表示参与最后一轮游戏的人数。

接下来  $n$  行，每行两个正整数，第  $i + 1$  行为  $a_i$  和  $b_i$  含义与题目描述一致。

### 输出格式

输出一行一个整数，表示在本轮游戏中，说谎人数的最少可能。

### 样例 #1

#### 样例输入 #1

```
3
2 0
0 2
2 2
```

#### 样例输出 #1

```
1
```

### 提示

【样例解释】

假设第 1 句话是真话，因为有 2 个人比他高，那么编号为 1 分数排名第 3；同理，假设第 2 句话是真话，2 号排名第 1，确定了 3 个人的排名为 2,3,1。

那么就是 3 在说谎，说谎人数为 1 人，并且可以通过枚举发现，说谎人数 1 人就是最小值。

【数据范围】

对于 10% 的数据满足： $n \leq 20$ ;

对于 30% 的数据满足： $n \leq 1000$ ;

对于 100% 的数据满足： $1 \leq n \leq 10^5, 0 \leq a_i + b_i \leq n$ 。

## T4 美食家

### 题目描述

美食家小明最近迷上了吃包子，于是上天降下来了  $n$  桶神奇的包子，编号为  $1, 2, \dots, n$ ，每桶包子的数量可以认为是无限的。

小明会从第 1 桶包子走到第  $n$  桶，这样总共走  $m$  趟，每一趟，都从第 1 桶走到第  $n$  桶。当他走到一桶包子前时，他就会在这里吃一次包子，有一件奇怪的事是，当小明吃一次包子时，所有包子桶中的包子都会减少。同时，小明发现了一个问题，就是他手太短了，只能拿到距离桶口  $\leq x$  范围内的包子，如果包子减少太多会导致他够不到包子。

一开始，第  $i$  桶包子的深度(桶口到包子的距离)为  $w_i$ ，每当小明吃一次包子，所有桶中包子的深度都会增加，其中，第  $i$  桶包子的深度增加  $a_i$ ，注意，小明吃包子可以认为的**瞬间的**，如果一开始够得着，但吃完后够不着，也视为一次成功的吃包子，包子深度也一样会增加。

小明想着知道他可以吃到多少次包子。

### 输入格式

第一行三个正整数  $n, m, x$ ，表示有  $n$  桶包子，小明会走总共  $m$  趟，能够到桶口与包子距离不超过  $x$  的包子。

第二行， $n$  个正整数，第  $i$  个数为第  $i$  桶包子的初始深度  $w_i$ 。

第三行， $n$  个正整数，第  $i$  个数为小明吃包子时第  $i$  桶包子增加的深度  $a_i$ 。

### 输出格式

一个正整数，为小明可以吃包子的次数。

### 样例 #1

#### 样例输入 #1

```
3 4 15
8 9 7
2 1 3
```

# 样例输出 #1

7

## 提示

【样例解释】

第一趟:

- 小明走到第 1 桶包子处, 成功吃了 1 次包子, 各桶深度变为 {10, 10, 10};
- 小明走到第 2 桶包子处, 成功吃了 1 次包子, 各桶深度变为 {12, 11, 13};
- 小明走到第 3 桶包子处, 成功吃了 1 次包子, 各桶深度变为 {14, 12, 16}。

第二趟:

- 小明走到第 1 桶包子处, 成功吃了 1 次包子, 各桶深度变为 {16, 13, 19};
- 小明走到第 2 桶包子处, 成功吃了 1 次包子, 各桶深度变为 {18, 14, 22};
- 小明走到第 3 桶包子处, 够不到包子, 各桶深度仍为 {18, 14, 22}。

第三趟:

- 小明走到第 1 桶包子处, 够不到包子, 各桶深度仍为{18, 14, 22};
- 小明走到第 2 桶包子处, 成功吃了 1 次包子, 各桶深度变为 {20, 15, 25};
- 小明走到第 3 桶包子处, 够不到包子, 各桶深度仍为{20, 15, 25}。

第四趟:

- 小明走到第 1 桶包子处, 够不到包子, 各桶深度仍为{20, 15, 25};
- 小明走到第 2 桶包子处, 成功吃了 1 次包子, 各桶深度变为{22, 16, 28};
- 小明走到第 3 桶包子处, 够不到包子, 各桶深度仍为{22, 16, 28}。

因此小明一共吃了 7 次包子。

【数据范围】

测试点	$n, m$	$x, a_i, w_i$	答案
1	$n \leq 10, m \leq 10^6$	$x, w_i \leq 2 \times 10^9$ $a_i \leq 200$	$\leq 10^3$
2	$n, m \leq 2 \times 10^4$		$\leq 10^6$
3	$n, m \leq 5 \times 10^4$		$\leq 10^7$
4			$\leq 5 \times 10^7$
5	$n, m \leq 10^5$		$\leq 10^8$
6			$\leq 2 \times 10^9$
7			
8			
9			
10			

