

# NOIP 模拟赛

WZMS

(请选手务必仔细阅读本页内容)

## 一、题目概况

中文题目名称	买邮票	数颜色	树上问题	字符串
子目录名	stamps	color	tree	string
可执行文件名	stamps	color	tree	string
输入文件名	stamps.in	color.in	tree.in	string.in
输出文件名	stamps.out	color.out	tree.out	string.out
每个测试点时限	1000ms	1000ms	3000ms	15000ms
内存上限	512M	512M	1G	1G
测试点数目	10	10	10	20
每个测试点分值	10	10	10	5
附加样例文件	有	有	有	有
结果比较方式	全文比较(过滤行末空格及文末回车)			
题目类型	传统	传统	传统	传统

## 二、提交源程序程序名

对于 C++ 语言	stamps.cpp	color.cpp	tree.cpp	string.cpp
-----------	------------	-----------	----------	------------

## 三、优化开关

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

### 注意事项:

1. 题目并不难, 请喧哗的同学不要大声 AK;
2. 文件名(程序名和输入输出文件名)必须使用英文小写;
3. C/C++ 中函数 main() 的返回类型必须是 int, 程序正常结束时的返回值必须是 0;
4. 保证各个题目的时间限制至少为标程运行时间的两倍;
5. 题目难度不一定升序排序;
6. 每道题目需单独建立对应子文件夹.

# 1 买邮票

(stamps.cpp)

## 1.1 问题描述

小 A 喜欢收集邮票。她现在在邮局购买一些新邮票。

世界上有  $N$  种不同种类的邮票：它们从 1 到  $N$  编号。然而，邮票不是单独出售的，它们必须以套装的形式购买。有  $M$  种不同的邮票套装可供选择，第  $i$  套装包含编号从  $L_i$  到  $R_i$  的邮票。同一张邮票可能会出现在多个套装中，也有可能一些邮票不存在于任何套装中。

所有套装的价格都是相同的。由于小 A 预算有限，她最多可以购买  $K$  种不同的套装。那么，爱丽丝最多可以获得多少种不同种类的邮票？

## 1.2 输入

从 `stamps.in` 输入数据。

第一行给定三个正整数  $N, M, K$ ，分别表示可用的不同种类的邮票数量、可用的邮票套装数量以及小 A 可以购买的最大邮票套装数量。

接下来  $M$  行，其中第  $i$  行表示第  $i$  个邮票套装，包含两个整数  $L_i$  和  $R_i$ ，它们表示该套装中可用邮票编号的包含范围。

## 1.3 输出

向 `stamps.out` 输出数据。

输出一个整数表示当前所求的答案。

## 1.4 输入输出样例 1

### 1.4.1 输入样例

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

### 1.4.2 输出样例

```
4
```

## 1.5 输入输出样例 1

### 1.5.1 输入样例

```
100 2 1
```

1 50  
90 100

### 1.5.2 输出样例

50

## 1.6 样例解释

在样例 1 中, 小 A 可以购买第一和第三个邮票套装, 它们包含了前四种邮票。请注意, 她获得了两份邮票 3, 但重要的是不同种类的邮票数量, 而不是每种邮票的数量。

在样例 2 中, 小 A 可以购买第一个邮票套装, 其中包含 50 种不同种类的邮票。

## 1.7 约定和数据范围

对所有测试点数据, 保证  $1 \leq K \leq M, 1 \leq N, M \leq 2000, 1 \leq L_i \leq R_i \leq N$ .

每个测试点的具体限制见下表:

测试点编号	$N$	$M$
1 ~ 2	$\leq 20$	$\leq 20$
3 ~ 6	$\leq 200$	$\leq 200$
7 ~ 10	$\leq 2000$	$\leq 2000$

## 2 数颜色

(color.cpp)

### 2.1 问题描述

温州是小 A 生活的城市，可以视为一个具有  $N$  个顶点和  $N$  个边的无向连通图。保证图中没有重复的边或自环。小 A 可以喷洒一种特殊类型的彩色墨水来装饰他们居住的道路。小 A 性情多变，所以他们经常改变道路的颜色来庆祝即将到来的 NOIP 活动。

NOIP 持续  $M$  天，每天小 A 将在一条道路上喷洒墨水，该道路上的颜色将被新墨水的颜色覆盖。在每天结束时，他们想知道温州有多少个不同颜色的区域。一个彩色区域是具有相同颜色的连接道路的集合，明确地说，如果两条道路具有相同的颜色并共享一个公共顶点，则它们属于同一个彩色区域。

### 2.2 输入

从 **color.in** 输入数据。

第一行包含两个整数  $N$  和  $M$ ，其中  $N$  是温州中的顶点和道路的数量， $M$  是 NOIP 持续的天数。

接下来的  $N$  行描述了顶点之间的道路。每行包含 3 个整数  $x, y$  和  $c$ ，表示第  $x$  个和第  $y$  个顶点之间有一条颜色为  $c$  的道路。

接下来的  $M$  行描述了每天的操作。每行包含 3 个整数  $x, y$  和  $c$ ，表示小 A 会在第  $x$  个和第  $y$  个顶点之间的道路上喷洒颜色为  $c$  的墨水。保证存在这样一条道路。

### 2.3 输出

向 **color.out** 输出数据。

输出共  $M$  行，其中第  $i$  行表示第  $i$  天时温州有多少个不同颜色的区域。

### 2.4 输入输出样例 1

#### 2.4.1 输入样例

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

### 2.4.2 输出样例

4  
3  
3

## 2.5 输入输出样例 2

### 2.5.1 输入样例

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

### 2.5.2 输出样例

2  
4  
2  
2

## 2.6 约定和数据范围

对所有测试点数据, 保证  $3 \leq N \leq 2 \times 10^5, 1 \leq M \leq 2 \times 10^5, 1 \leq x, y, c \leq n$ .  
每个测试点的具体限制见下表:

测试点编号	$N$	$M$
1 ~ 3	$\leq 2000$	$\leq 2000$
4	无	$\leq 20$
5	$\leq 20$	无
6 ~ 7	$\leq 100000$	$\leq 100000$
8 ~ 10	无	无

## 3 树上问题

(tree.cpp)

### 3.1 问题描述

你有一棵包含  $N$  个顶点的有根树。顶点 1 是根，其他  $N - 1$  个顶点都有且仅有一条入边。每个顶点  $i$  有  $C_i$  个人居住。

最初，所有边都是蓝色的。你可以将一条蓝色路径成一条红色边。具体来说，当有  $k$  条蓝色边时， $(a_1, a_2), (a_2, a_3), \dots, (a_k, a_{k+1})$ ，你可以将它们替换为一条红色边， $(a_1, a_{k+1})$ 。你可以执行这个操作任意次数。

你的目标是防止人员之间的接触，所以希望你尽量减少总接触次数。

总接触次数是指那些生活在不同顶点的人员对  $(A, B)$  的数目，其中  $A$  和  $B$  可以通过边（无论颜色如何）互相访问。注意，边是有向的。

请找出在树上进行一些（可能为零次）操作后可以实现的最小总接触次数。

### 3.2 输入

从 **tree.in** 输入数据。

第一行包含一个整数  $N$ ，表示顶点的数量。

接下来一行包含  $N - 1$  个整数， $P_2, P_3, \dots, P_N$ ，表示顶点  $i$  有一条来自顶点  $P_i$  的入边。这些数字描述了一个以顶点 1 为根的有根树。请记住这些边是有向的。

接下来一行包含  $N$  个整数， $C_1, C_2, \dots, C_N$ ，表示每个顶点中的人数。

### 3.3 输出

向 **tree.out** 输出数据。

输出一个整数，表示最小总接触次数。

### 3.4 输入输出样例 1

#### 3.4.1 输入样例

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

#### 3.4.2 输出样例

```
10
```

### 3.5 约定和数据范围

对所有测试点数据, 保证  $1 \leq N \leq 2 \times 10^5, 1 \leq P_i \leq N, 1 \leq C_i \leq 1000000$

每个测试点的具体限制见下表:

测试点编号	$N$	特殊性质
1 ~ 4	$\leq 2000$	无
5	无	树的形态是一条链
6 ~ 7	$\leq 100000$	无
8 ~ 10	无	无

## 4 字符串

(string.cpp)

### 4.1 问题描述

选择一个长度为  $N$  的由小写英文字母组成的字符串。例如，字符串 `aba`。

找出字符串从位置  $0, 1, \dots, N-1$  开始的所有后缀，并按字典序排序。在范例中，我们得到的后缀是 `aba, ba, a`。按照字典序排序之后是 `a, aba, ba`。

顺序遍历排序后的后缀，记下每个后缀在原串中的开始位置。在范例中，排在排序列表第一个的后缀 `a` 是由从下标为 2 的位置开始，后缀 `aba` 从位置 0 开始，后缀 `ba` 开始从位置 1 开始。所以我们会得到数组  $[2, 0, 1]$ 。

回到原来的字符串，对于每一个位置，或两个连续位置的中点，将其作为回文中心求出最大的回文半径，并将其写下。在范例中我们可以得到数组  $[1, 0, 3, 0, 1]$ 。

我们现在的任务是通过给定两个数组，求是否能够构造一个字符串使得他的后缀排名和最大回文半径满足给定的条件。

如果可以，那么输出字典序最小的字符串。否则，输出 `-1`。

通俗的说，就是通过给出后缀数组以及 manacher 中得到的回文半径数组来构造一个字符串。

### 4.2 输入

从 `string.in` 输入数据。

下面的描述下标都是从 0 开始。

第一行一个整数  $T$ ，表示数据组数。

对于每一组数据，第一行一个整数  $n$ ，表示长度。字符串下标从 0 开始。

第二行  $n$  个整数，表示给定的后缀数组。

第三行  $2n-1$  个整数，表示最大回文半径数组，其中第  $2i$  个整数表示以第  $i$  个字符为中心的最长回文串长度， $2i+1$  个整数表示以第  $i$  个字符和第  $i+1$  个字符之间为中心的最大回文串半径长度。

### 4.3 输出

向 `string.out` 输出数据。

有解则输出字典序最小且满足条件的串，否则输出 `-1`。

### 4.4 输入输出样例

详见下发文件中 `string` 文件夹中的内容。

- ex1 样例
- ex2 对应 5 的测试点。
- ex3 对应 8 的测试点。



- ex4 对应 16 ~ 20 的测试点。

#### 4.5 约定和数据范围

对于所有测试点， $T \leq 5, n \leq 10^6$ ，保证输入都是  $[0, n]$ 。

每个测试点的具体限制见下表：

测试点编号	$n$	特殊性质 A	特殊性质 B	特殊性质 C
1 ~ 3	$\leq 8$	无	无	无
4	$\leq 100$	有	无	无
5	$\leq 100$	无	有	无
6	$\leq 100$	无	无	有
7	$\leq 100$	无	无	无
8	$\leq 3000$	有	无	无
9	$\leq 3000$	无	有	无
10	$\leq 3000$	无	无	有
11 ~ 12	$\leq 3000$	无	无	无
13 ~ 14	$\leq 10^6$	有	无	无
15	$\leq 10^6$	无	有	无
16 ~ 20	$\leq 10^6$	无	无	无

特殊性质 A：保证后缀数组形如  $n-1, n-2, \dots, 0$ 。

特殊性质 B：保证最大回文半径都是 0 或 1。

特殊性质 C：保证答案要么无解，要么只含两个字符。