# JSTSC 2016

## 第二轮第二试

**竞赛时间：2016 年 5 月 18 日 8:00-13:00**

| 题目名称 | 病毒感染 | 反质数序列 | 炸弹攻击 2 |
|---|---|---|---|
| 可执行文件名 | virus.exe | prime.exe | attack2.exe |
| 输入文件名 | virus.in | prime.in | attack2.in |
| 输出文件名 | virus.out | prime.out | attack2.out |
| 每个测试点时限 | 1 秒 | 1 秒 | 2 秒 |
| 内存限制 | 512M | 512M | 512M |
| 测试点数目 | 10 | 10 | 10 |
| 每个测试点分值 | 10 | 10 | 10 |
| 是否有部分分 | 否 | 否 | 否 |
| 题目类型 | 传统型 | 传统型 | 传统型 |

提交源程序须加后缀

| | | | |
|---|---|---|---|
| 对于 Pascal 语言 | virus.pas | prime.pas | attack2.pas |
| 对于 C　　语言 | virus.c | prime.c | attack2.c |
| 对于 C++　语言 | virus.cpp | prime.cpp | attack2.cpp |

**注意：最终测试时不打开任何优化开关**

# 病毒感染

## 【故事背景】

JOSI 的边陲小镇爆发了严重的 Jebola 病毒疫情，大批群众感染生命垂危。计算机科学家 JYY 采用最新的算法紧急研制出了 Jebola 疫苗，并火速前往灾区救治患者。

## 【问题描述】

一共有 $N$ 个小镇爆发了 Jebola 疫情。这些小镇由于地处边陲，仅仅通过一条长直公路连接。方便起见我们将这些小镇按照公路连接顺序有 1 编号到 $N$。JYY 会在第一天一早抵达 1 号小镇。

一开始在 $i$ 号小镇，有$a_i$名患者感染了 Jebola 病毒。

每一天 JYY 可以选择：

1. 花费一天时间彻底治愈 JYY 目前所在的村庄的所有 Jebola 患者。这一天不会有任何患者死去。
2. 花费一天的时间前往一个相邻的村庄。

当一天开始时，如果一个村庄里有 $k$ 个 Jebola 患者，那么这一天结束时，这$k$个患者会感染另外$k$个这个村子里的健康村民并死去。所以对于 $i$ 号村庄，只要这个村庄没有被 JYY 彻底消灭疫情，那么每一天都会有$a_i$个村民死去。

JYY 希望采用措施使得疫情被整体消灭时，总共死去的村名数量尽量少。

为了达成这一目标，JYY 有时会选择抵达一个村庄但是并不对村民进行施救。这样的行为如果不加限制，往往会造成更加严重的后果。

试想这样的情形：假设当 JYY 第一次抵达村庄 $i$，未作救治并直接前往了另一个村庄。那么由于 $i$ 村庄的人们求生心切，一旦当 JYY 朝向靠近 $i$ 村庄的方向前行时，$i$ 村庄的村民就会以为 JYY 是来救他们了，而产生巨大的期望。之后倘若 JYY 再次掉头朝着远离 $i$ 村庄的方向行进，那么 $i$ 村庄的村民就会因为巨大的失落而产生绝望的情绪。

为了避免这种情况，JYY 对他的行程做了如下规定：

假设 JYY 进入 $i$ 村庄并在第二天立即离开（村庄 $i$ 的疫情并未治愈）。如果在之后的某一天，JYY 从村庄 $j$ 前往村庄 $k$，并满足$|k-i|<|k-j|$。那么在之后的日子里 JYY 只能朝着 $i$ 村庄前进直到抵达 $i$ 村庄并立即治愈该村的患者。在前往 $i$ 村庄的过程中，JYY 可以选择将途经村庄的疫情治愈。

比如，如果 JYY 有如下行程：

第一天：从村庄 1 前往村庄 2；第二天：从村庄 2 前往村庄 3；第三天：治愈村庄 3；第四天：前往村庄 2。

此时 JYY 对于之后三天的行程只有唯一一种选择：

第五天：治愈村庄 2；第六天：前往村庄 1；第七天：治愈村庄 1。

JYY 想知道在治愈所有村庄之前，至少会有多少村民因 Jebola 死去。

【输入格式】

从文件 ***virus.in*** 中读入数据。

输入第一行包含一个正整数 $N$。

接下来一行包含 $N$ 个整数，分别为$a_1, a_2, \ldots, a_N$。

【输出格式】

输出到文件 ***virus.out*** 中。

输出一行一个整数，表示最优行程安排下会死去的村民数量。

【样例输入】

```
6
40 200 1 300 2 10
```

【样例输出】

```
1950
```

【样例解释】

我们用 C($k$)表示治愈 $k$ 号村庄，$i$→$j$ 表示从村庄 $i$ 前进到村庄 $j$，用分号分隔每一天的行程安排，那么样例中的最优策略为：

1→2; C(2); 2→3; 3→4; C(4); 4→3; C(3); 3→2; 2→1; C(1); 1→2; 2→3; 3→4; 4→5; 5→6; C(6); 6→5; C(5)。

整个过程耗时 18 天。

【数据规模与约定】

对于 10%的数据满足$N \leq 10$；

对于 30%的数据满足$N \leq 20$；

对于 50%的数据满足$N \leq 60$；

对于 100%的数据满足$1 \leq N \leq 3000, 1 \leq a_i \leq 10^9$。

# 反质数序列

## 【故事背景】

由于发现太多以质数为条件的问题，计算机科学家JYY开始对质数厌烦了。

## 【问题描述】

对于一个长度为$L \geq 2$的序列$X: x_1, x_2, \ldots, x_L$，如果满足对于任意$1 \leq i < j \leq L$，均有$x_i + x_j$不为质数，则 JYY 认为序列 $X$ 是一个“反质数序列”。

JYY 有一个长度为 $N$ 的序列$A: a_1, a_2, \ldots, a_N$，他希望从中选出一个包含元素最多的***子序列***，使得这个子序列是一个反质数序列。

## 【输入格式】

从文件 ***prime.in*** 中读入数据。

输入第一行包含一个正整数 $N$。

接下来一行包含 $N$ 个正整数，依次描述$a_1, a_2, \ldots, a_N$。

## 【输出格式】

输出到文件 ***prime.out*** 中。

输出一行一个整数，表示最长反质数子序列的长度。输入保证存在反质数子序列。

## 【样例输入】

```
6
1 2 2 3 4 10
```

## 【样例输出】

```
4
```

## 【数据规模与约定】

对于 10%的数据满足$N \leq 10$；

对于 40%的数据满足$N \leq 150$；

对于 80%的数据满足$N \leq 1000$；

对于 100%的数据满足$2 \leq N \leq 3000, 1 \leq a_i \leq 10^5$。

# 炸弹攻击 2

## 【故事背景】

还记得那款题为“炸弹攻击”的塔防游戏吗？这款游戏出了续作，炸弹的威力大大加强了。

## 【问题描述】

游戏的地图是一个 2 维平面。JYY 的阵地位于 $x$ 轴下方，而所有的敌人目前都位于 $x$ 轴上方。

在 JYY 的阵地中有建有 $T$ 个激光塔和 $S$ 个发射源。其中第 $i$ 个防御塔$T_i$的坐标为$(tx_i, ty_i)$，第 $i$ 个发射源$S_i$的坐标为$(sx_i, sy_i)$。

地图上有 $D$ 个敌人，第 $i$ 个敌人$D_i$的坐标为$(dx_i, dy_i)$。

两座激光塔可以相互连接形成能量墙。发射源***朝向敌人***发出的能量如果穿过了能量墙，可以得到巨大的加强而变为“超级射线”并瞬间消灭敌人。

JYY 想知道他有多少种可以可以发出超级射线的攻击方案。

具体来说，一个可以发出超级射线的攻击方案为一个由四个点组成的集合：$\{T_i, T_j, S_k, D_l\}$，满足$1 \le i < j < T, 1 \le k \le S, 1 \le l \le D$ 并且线段$T_iT_j$和线段$S_kD_l$相交。

游戏设定保证在这$T + D + S$个点中，不存在重点也不存在三点共线。

## 【输入格式】

从文件 ***attack2.in*** 中读入数据。

第一行包含一个正整数 $D$。

接下来 $D$ 行，每行包含两个整数$dx_i, dy_i$ $(dy_i > 0)$，表示一个敌人的坐标。

第 $D$+1 行包含一个整数 $S$。

接下来 $S$ 行，每行包含两个整数$sx_i, sy_i$ $(sy_i < 0)$，表示一个发射源的坐标。

第 $D$+$S$+2 行包含一个整数 $T$。

接下来 $T$ 行，每行包含两个整数$tx_i, ty_i$ $(ty_i < 0)$，表示一个激光塔的坐标。

## 【输出格式】

输出到文件 ***attack2.out*** 中。

输出一行一个整数，可以发出超级射线的攻击方案个数。

## 【输入样例】

```
3
1 12
10 30
```

```
30 10
1
10 -10
4
2 -11
9 -1
11 -1
15 -14
```

【输出样例】

```
7
```

【数据规模】

对于 20%的数据满足 $D,S,T \leq 30$。

对于 50%的数据满足 $D,S,T \leq 150$。

对于 100%的数据满足$1 \leq D,S,T \leq 800$，所有坐标绝对值不超过$10^9$。