

# 2024 年计算机编程素养提升交流活动

## 拔尖班

### 图结构

时间：2024 年 7 月 4 日 08:00 ~ 11:30

题目名称	矿石病	接龙	移动	旅行
题目类型	传统型	传统型	传统型	传统型
可执行文件名	oripathy.exe	play.exe	move.exe	travel.exe
输入文件名	oripathy.in	play.in	move.in	travel.in
输出文件名	oripathy.out	play.out	move.out	travel.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	10	10	13	10
测试点是否等分	否	否	否	否

提交源程序文件名

对于 C++ 语言	oripathy.cpp	play.cpp	move.cpp	travel.cpp
-----------	--------------	----------	----------	------------

编译选项

对于 C++ 语言	-O2 -std=c++14 -Wl,--stack=536870912
-----------	--------------------------------------

# 矿石病 (oripathy)

## 【题目描述】

现有  $n$  个感染者，感染程度分别为  $a_1 \dots a_n$ 。兔兔需要将他们分成两组以避免矿石病的扩散。

假设兔兔已将编号划分为  $S, T$  两个集合，矿石病不会扩散当且仅当

$$\gcd(\prod_{i \in S} a_i, \prod_{i \in T} a_i) = 1$$

由于医疗资源紧张，兔兔不希望有一组为空。即应有

- $S \cup T = \{1, \dots, n\}$
- $S \cap T = \emptyset$
- $S, T \neq \emptyset$

兔兔想知道划分方案数对  $10^9 + 7$  取模的结果。

## 【输入格式】

从文件 *oripathy.in* 中读入数据。  
第一行为一个非负整数  $t$ ，表示数据组数。  
接下来  $t$  组数据。每组数据的第一行为一个正整数  $n$ 。  
第二行为  $n$  个正整数  $a_1 \dots a_n$ 。

## 【输出格式】

输出到文件 *oripathy.out* 中。  
输出  $t$  行。每行一个非负整数，为划分方案数对  $10^9 + 7$  取模的结果。

## 【样例 1 输入】

```
1 1
2 4
3 9 6 2 5
```

## 【样例 1 输出】

```
1 2
```

## 【样例 1 解释】

可行的划分方案为  $S = \{9, 6, 2\}, T = \{5\}$  与  $S = \{5\}, T = \{9, 6, 2\}$ 。

【样例 2】

见选手目录下的 *oripathy/oripathy2.in* 与 *oripathy/oripathy2.ans*。

【子任务】

对于所有测试数据，保证  $1 \leq t \leq 5, 1 \leq n \leq 10^5, 1 \leq a_i \leq 10^6$ 。

测试点编号	分值	$n \leq$	$a_i \leq$
1	5	15	15
2			
3			
4			
5	15	$10^3$	$10^3$
6			
7	10	$10^5$	1
8			2
9	15		$10^6$
10			

## 接龙 (play)

### 【题目描述】

兔兔想和博士玩接龙游戏。给定一个含有  $n$  个单词的词典，接龙的规则如下：

- 兔兔先说，博士后说
- 从第二个单词起，所选单词的前三个字母必须等于上一个单词的后三个字母
- 单词允许重复使用，字母区分大小写
- 当无合法的单词可以接龙时，判当前玩家为负

兔兔希望知道，如果她首先说出第  $i$  个单词，最终的赢家会是谁。当然，兔兔和博士都会选择对自己最优的方案，因此可能会出现平局。

### 【输入格式】

从文件 *play.in* 中读入数据。

第一行为一个非负整数  $n$ ，表示词典大小。

接下来  $n$  行。每行为一个单词。

### 【输出格式】

输出到文件 *play.out* 中。

输出  $n$  行，每行一个字符串。兔兔赢则输出 **Amiya**，博士赢则输出 **Doctor**，平局则输出 **Draw**。

### 【样例 1 输入】

```
1 3
2 abcd
3 bcda
4 ada
```

### 【样例 1 输出】

```
1 Doctor
2 Amiya
3 Draw
```

【样例 1 解释】

- 兔兔先说 `abcd`，博士接着说 `bcda`，于是兔兔接下来无话可说，博士赢
- 兔兔先说 `bcda`，博士无话可说，兔兔赢
- 兔兔先说 `ada`，博士接着说 `ada`，陷入循环，平局

【样例 2】

见选手目录下的 `play/play2.in` 与 `play/play2.ans`。

【子任务】

对于所有测试数据，保证  $1 \leq n \leq 2 \times 10^5$ ，单词长度为  $[3, 8]$  之间的整数。

测试点编号	分值	$n \leq$	特殊说明
1	5	1	无
2		5	
3	10	$10^4$	
4			
5	5		
6			
7	15	$2 \times 10^5$	无
8			
9			
10			

# 移动 (move)

## 【题目描述】

将泰拉大陆看作一个由  $V$  个城市、 $E$  条道路构成的无向图。有  $N$  位干员分散在图上各城市中，其在城市  $u_i, v_i$  间移动的耗时均为  $t_i$ 。 $T = 0$  时刻，干员们开始各自移动。

为了这片大地的和平，老猫希望他们以某种方式移动，从而在最短的时间内使得  $K$  个城市中存在干员。

老猫无所不知，她完全知道该如何安排，所以她只需要博士计算出这个理论所需时间。

## 【输入格式】

从文件 `move.in` 中读入数据。

第一行四个非负整数  $V, E, N, K$ ，分别表示城市数量、道路数量、干员数量与老猫的要求。

接下来一行  $N$  个正整数，第  $i$  个数表示第  $i$  位干员所在城市的编号。

接下来  $E$  行，每行三个正整数  $u_i, v_i, t_i$ ，表示城市  $u_i, v_i$  间存在一条需要耗时  $t_i$  通过的道路。

## 【输出格式】

输出到文件 `move.out` 中。

输出一行。为一个整数  $T$ ，表示使得  $K$  个城市存在干员需要的最短时间。如果老猫的要求无法满足，输出  $-1$ 。

## 【样例 1 输入】

```
1 6 7 5 4
2 5 5 2 2 5
3 1 3 3
4 1 5 2
5 1 6 5
6 2 5 4
7 2 6 7
8 3 4 11
9 3 5 3
```

**【样例 1 输出】**

1 3

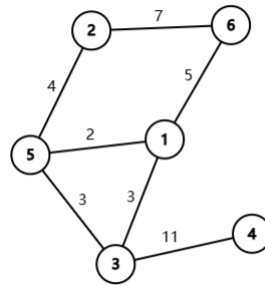
**【样例 1 解释】**

图 1: 样例中的图

初始时，干员 1、2、5 在城市 5，干员 3、4 在城市 2。

最优的移动方式为：

- 干员 1 去城市 3，耗时 3
- 干员 2 去城市 1，耗时 2

于是 3 单位时间后，城市 1、2、3、5 均有干员存在，符合  $K = 4$  的要求。可以证明不存在比 3 更小的答案。

**【样例 2】**

见选手目录下的 *move/move2.in* 与 *move/move2.ans*。

**【子任务】**

对于所有测试数据，保证  $1 \leq V \leq 600, 1 \leq E \leq 2 \times 10^4, 1 \leq N \leq \min(V, 200), 1 \leq K \leq N, 1 \leq t_i \leq 10^4$ 。

测试点编号	分值	$V \leq$	$E \leq$	$N \leq$	$K \leq$	特殊说明		
1	5	100	2000	50	50	无		
2								
3								
4								
5								
6	10	300	$10^4$	200	200	$K = N$		
7								
8		600	$2 \times 10^4$					
9								
10								
11								
12								
13								



# 旅行 (travel)

## 【题目描述】

博士的计算符合老猫的预期，她非常满意，因此准许你和兔兔在这片大地上旅行一段时间。

同样将泰拉大陆看作一个由  $V$  个城市、 $E$  条道路构成的无向图，在城市  $u_i, v_i$  间移动的耗时为  $t_i$ 。确定了旅途的起点与终点后，二人将会沿着耗时最短的路径移动。

兔兔制定了  $q$  个旅行方案（其实只有起点和终点），她希望博士能计算出各个方案的耗时。

## 【输入格式】

从文件 *travel.in* 中读入数据。

第一行两个正整数  $V, E$ ，为城市数量与道路数量。

接下来  $E$  行，每行三个正整数  $u_i, v_i, t_i$ ，表示城市  $u_i, v_i$  间存在一条需要耗时  $t_i$  通过的道路。

接下来一行为一个正整数  $q$ ，为询问数量。

接下来  $q$  行，每行两个正整数  $s_i, t_i$ ，表示每个方案的起点与终点。

## 【输出格式】

输出到文件 *travel.out* 中。

输出  $q$  行。每行一个整数  $T$ ，第  $i$  行为执行第  $i$  个方案的用时。

## 【样例 1 输入】

```
1 3 3
2 1 2 3
3 2 3 1
4 3 1 5
5 3
6 1 2
7 1 3
8 2 3
```

## 【样例 1 输出】

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

【样例 1 解释】

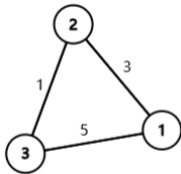


图 2: 样例中的图

- 对于第一个方案，路径  $1 \rightarrow 2$  耗时最短，为 3。
- 对于第二个方案，路径  $1 \rightarrow 2 \rightarrow 3$  耗时最短，为 4。
- 对于第三个方案，路径  $2 \rightarrow 3$  耗时最短，为 1。

【样例 2】

见选手目录下的 `travel/travel2.in` 与 `travel/travel2.ans`。

【子任务】

对于所有测试数据，保证  $1 \leq V, E, q \leq 10^5, E - V \leq 20, 1 \leq t_i \leq 10^9$ ，图连通且不存在重边与自环。

测试点编号	分值	$V \leq$	$q \leq$
1	10	300	$10^5$
2			
3	5	$10^3$	100
4		$10^4$	
5	15	$10^3$	$10^5$
6			
7	10	$10^5$	
8			
9			
10			