

# 基础题巩固 2

请注意题目难度可能与顺序无关。

题目难度不大，为锻炼细节能力，本次题目不设部分分。

## 1 Problem A

时空限制：20 sec / 512 MB

### 1.1 问题描述

Colin 有  $n$  个数字  $a_1, \dots, a_n$ ，想让你统计  $\gcd(a_i, a_j) = a_i \oplus a_j$  的有序对  $(i, j)$  ( $1 \leq i < j \leq n$ ) 的个数。

其中  $\oplus$  表示按位异或运算。

### 1.2 输入格式

第一行一个整数  $T$  ( $T \leq 20$ )，代表数据组数，对于每组数据：

第一行一个正整数  $n$  ( $1 \leq n \leq 2 \times 10^6$ )，代表数列的长度。保证所有数据  $\sum n \leq 3 \times 10^7$ 。

第二行  $n$  个正整数  $a_i$  ( $1 \leq a_i \leq 10^6$ )，代表给定的  $n$  个数字。

### 1.3 输出格式

对于每组输出一行一个整数，代表答案。

### 1.4 输入样例

1		1
2		4
3		2 3 4 3

### 1.5 输出样例

1		2
---	--	---

## 2 Problem B

时空限制：1.5 sec / 256 MB

### 2.1 问题描述

Colin 有一个数列  $\{a\}$ ，他想把序列变成不降子序列，依次做如下三个操作:

- 删除任意长度的前缀（可以为 0），没有代价。
- 删除任意长度的后缀（可以为 0），没有代价。
- 删除序列中若干位置，每删除一个位置代价为 1。

Colin 想要保证最后剩下的数列中数字单调不降的前提下，最大化数列的长度，再最小化操作的代价。

### 2.2 输入格式

第一行一个整数  $T$  ( $T \leq 10^5$ )，代表数据组数，对于每组数据:

第一行一个正整数  $N$  ( $N \leq 10^5$ )，代表数列的长度。保证所有数据  $\sum N \leq 10^6$ 。

第二行  $N$  个正整数  $a_i$  ( $1 \leq a_i \leq 10^9$ )，代表输入的数列。

### 2.3 输出格式

对于每组数据输出一行两个整数  $x\ y$ ，代表剩余数列的最长长度，和操作代价的最小值。

### 2.4 输入样例

```
1 | 1
2 | 10
3 | 4 3 2 1 5 6 3 3 1 3
```

### 2.5 输出样例

```
1 | 4 3
```

### 3 Problem C

时空限制：10 sec / 1024 MB

#### 3.1 问题描述

Colin 有一个  $n \times m$  的矩阵  $A$ ，请你维护，支持：

- 1 x y v：将  $A[x][y]$  改为  $v$ ；
- 2 x1 y1 x2 y2：查询以  $(x1, y1)$  为左上角， $(x2, y2)$  为右下角的子矩形  $A'$  中，最大的子矩形和。  
也就是说，查询所有被  $A'$  完全盖住的子矩形里，矩形和的最大值（不能为空）。

#### 3.2 输入格式

第一行两个正整数  $n, m$  ( $1 \leq n \leq 10, 1 \leq m \leq 10^5$ )，描述矩阵大小。

接下来  $n$  行，每行  $m$  个整数  $A[i][j]$  ( $-10^9 \leq A[i][j] \leq 10^9$ ) 描述矩阵初始每个位置的值。

接下来一行一个整数  $q$  ( $1 \leq q \leq 1000$ )，描述操作次数。

接下来  $q$  行，每行的输入表示上述两个操作之一，输入格式同上。

数据范围： $1 \leq x \leq n, 1 \leq x1 \leq x2 \leq n, 1 \leq y \leq m, 1 \leq y1 \leq y2 \leq m, -10^9 \leq x \leq 10^9$ 。

#### 3.3 输出格式

对于每个 2 操作，输出一行一个整数，代表查询的结果。

#### 3.4 输入样例

```
1 | 2 3
2 | 3 5 2
3 | -1 -3 -1
4 | 3
5 | 2 1 1 2 3
6 | 1 2 2 3
7 | 2 1 1 2 2
```

#### 3.5 输出样例

```
1 | 10
2 | 10
```

## 4 Problem D

时空限制：1 sec / 1024 MB

### 4.1 问题描述

Colin 维护了一个树状数组  $c[1 \dots n]$ ，开始时数组中所有位置都是 0，其中更新函数的代码如下：

```
1 void update(int pos, double val) {
2     while (pos <= n) {
3         c[pos] += val;
4         pos += pos & (-pos); // lowbit(pos)
5     }
6 }
```

运行过一段程序之后，Colin 得到了经过若干次更新的  $c$  数组。

但是他一不小心把源代码删了，而且因为某种不可抗力只能知道  $c[i]$  是否为 0。

现在 Colin 想问你，在最好的情况下，最少需要再进行多少次 update 操作才能将  $c[i]$  全部变为 0？

在本题中，认为更新操作中的  $val$  可以选取任意实数。

### 4.2 输入格式

第一行一个正整数  $T$  ( $1 \leq T \leq 10^5$ )，代表数据组数，对于每组数据：

第一行一个正整数  $n$  ( $1 \leq n \leq 10^5$ )，代表数组的大小，保证所有数据  $\sum n \leq 10^6$ 。

第二行一个  $n$  位 01 串，从左数第  $i$  位是 0 代表最终  $c[i]$  为 0，反之不为 0。

### 4.3 输出格式

对于每组数据输出一行一个整数，代表所需的最小操作数。

### 4.4 输入样例

```
1 3
2 5
3 10110
4 5
5 00000
6 5
7 11111
```

### 4.5 输出样例

```
1 3
2 0
3 3
```

# 5 Problem E

时空限制：1.5 sec / 256 MB

## 5.1 问题描述

Colin 在玩一个氪金游戏，玩家的战斗力指标有  $m$  个，开始的时候全部为 0。

游戏为了吸引玩家开服第一个月每天都有登陆礼包，礼包分为免费礼包和氪金礼包两种，每天只能选一个领取。

每个礼包的效果是对每个指标都会有一定的提升，具体的：

- 第  $i$  天的免费礼包对第  $j$  个指标的增益是  $x_{i,j}$ ，也就是会让第  $j$  个指标加上  $x_{i,j}$
- 第  $i$  天的氪金礼包对第  $j$  个指标的增益是  $y_{i,j}$ ，也就是会让第  $j$  个指标加上  $y_{i,j}$

如果 Colin 当天想要氪金礼包，他就需要氪金一次。

无良开发商对战斗力做出了苛刻的要求：只有第  $j$  个指标恰好为  $b_j$  时，玩家的战斗力才是最强的。

作为一名强迫症玩家，Colin 自然想要最强的战斗力，但他手头有没有很多钱，所以：

假设只能通过领取礼包提升指标，并且一定要领取完这个月所有天的礼包（每天二选一），Colin 为了达到最强战斗力至少需要氪金多少次？

## 5.2 输入格式

第一行一个正整数  $T$  ( $1 \leq T \leq 30$ )，代表数据组数，对于每组数据：

第一行两个正整数  $n, m$  ( $1 \leq n, m \leq 30$ )，代表开服第一个月的天数和战斗力指标的个数。

第二行  $m$  个整数  $b_1, b_2, \dots, b_m$  ( $0 \leq b_i \leq 2000$ )，第  $j$  个数字代表无良开发商对第  $j$  个指标的要求。

接下来  $n$  行，每行  $2 \times m$  个整数，第  $i$  行依次为  $x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2}, \dots, x_{i,m}, y_{i,m}$  ( $0 \leq x_{i,j}, y_{i,j} \leq 100$ )

## 5.3 输出格式

对于每组数据输出一行，如果存在可行方案，输出一个整数代表最少氪金次数；否则输出 impossible。

## 5.4 输入样例

1		1
2		3 2
3		10 5
4		3 4 2 0
5		3 1 1 4
6		1 4 4 2

## 5.5 输出样例

1		1
---	--	---

