



University

---

**BCPC 2018** The 14th Beihang University  
Collegiate Programming Contest

---

## 比赛题目

- A. wjj 的分手厨房
- B. zzh 与子序列
- C. 未闻题名
- D. 区间 mex
- E. 唧个的 Scramble String
- F. 构造题
- G. 进化吧亚古兽!!!
- H. Ange 的线性代数题
- I. 击鼓传花

*比赛开始前请勿翻阅*



## Problem A. wjj 的分手厨房

Input file: *standard input*

Output file: *standard output*

Time limit: 1 second

Memory limit: 256 mebibytes

现充 wjj 最近沉迷一个叫做分手厨房的游戏。这一天，wjj 在训练摸鱼的时候打起了瞌睡，他梦到了自己成为了一名火车上的厨师。

火车的餐厅里有  $n$  个客人，菜单上共有  $m$  种菜，第  $i$  种菜需要  $t_i$  的制作时间。每个客人都在时刻 0 到达餐厅，且下了不同的订单，其中第  $i$  个客人订了  $k_i$  道菜。这些客人都有一个共同的习惯，就是只有当他们的菜上齐之后他们才会开始吃饭。

众所周知 wjj 是一个游戏菜鸡，他糟糕的做菜速度让客人们急不可耐。所幸的是 wjj 在梦中拥有一个异次元大锅，这个锅的体积是无限大的，也就是说他可以同时制作出任意多道同一种菜肴。

当所有客人都开始吃饭之后，wjj 就可以在梦中愉快的摸鱼了。你能帮聪明的 wjj 算算他最早什么时刻可以开始摸鱼吗？

### Input

第一行包含两个整数  $n, m$ ，满足  $1 \leq n, m \leq 1000$ 。

第二行包含  $m$  个正整数  $t_1, t_2, \dots, t_m$ ，满足  $1 \leq t_i \leq 10\,000$ 。

接下来  $n$  行，其中第  $i$  行的第一个数为  $k_i$ ，满足  $1 \leq k_i \leq 10$ ，随后包含  $k_i$  个正整数，依次代表这  $k_i$  道菜的编号。

### Output

一个数，即 wjj 最早开始摸鱼的时刻。

### Example

standard input	standard output
3 2 1 2 1 1 2 1 2 2 2 1	3

## Problem B. zzh 与子序列

Input file: *standard input*

Output file: *standard output*

Time limit: 3 seconds

Memory limit: 128 mebibytes

宝可梦文是宝可梦世界通行的语言，这种语言一共有  $\Sigma$  种字母，我们用整数  $1, 2, \dots, \Sigma$  来标记它们。

一天，zzh 和皮卡丘在冒险中发现了一个宝可梦文字符串  $s$ 。他们希望知道  $s$  字典序第  $k$  小的非空子序列是什么，请你帮他们找出这个子序列。

**注1:** 设有两长度相同的字符串  $s_1s_2\cdots s_n$  和  $t_1t_2\cdots t_n$ ， $s$  的字典序小于  $t$  当且仅当  $s$  第一个与  $t$  不同的位置  $i$  满足  $s_i < t_i$ 。

若两字符串长度不同，将较短的字符串补上空字符（小于其它任何字符）再进行比较。

**注2:** 字符串的子序列是从原始字符串通过去除某些元素但不破坏余下元素的相对位置（在前或在后）而形成的新字符串。

非空子序列是指余下的元素不能为空集。

**注意:** 重复出现的子序列需要重复计算，详见 Note。

### Input

第一行包含一个正整数  $T$  ( $1 \leq T \leq 1000$ )，表示有  $T$  组测试数据。

接下来依次给出每组测试数据。对于每组测试数据：

第一行包含三个整数  $n$  ( $1 \leq n \leq 5000$ )， $\Sigma$  ( $1 \leq \Sigma \leq 5000$ ) 和  $k$  ( $1 \leq k \leq \min\{2^n - 1, 10^{18}\}$ )，其中  $n$  表示字符串  $s$  的长度， $\Sigma$  和  $k$  的含义见题目描述。

第二行包含  $n$  个整数，第  $i$  个整数为  $s_i$  ( $1 \leq s_i \leq \Sigma$ )，表示  $s$  的第  $i$  个字符。

保证不超过 3 组测试数据满足  $n > 100$ 。

### Output

对于每组数据，输出一行，包含若干个整数，表示该组数据的答案。

### Example

standard input	standard output
3	1 3
2 3 2	1
1 3	1 2 1
1 1 1	
1	
3 2 5	
1 2 1	

## Note

对于第一组数据，子序列依次为  $[1]$ ,  $[1, 3]$ ,  $[3]$ 。

对于第二组数据，子序列为  $[1]$ 。

对于第三组数据，子序列依次为  $[1]$ ,  $[1]$ ,  $[1, 1]$ ,  $[1, 2]$ ,  $[1, 2, 1]$ ,  $[2]$ ,  $[2, 1]$ 。

## Problem C. 未闻题名

Input file: *standard input*  
 Output file: *standard output*  
 Time limit: 2 seconds  
 Memory limit: 512 mebibytes

我们今天还没有等来一个新题.jpg

L 头子

随着校赛的临近，出题组的 L 头子从 2 月开始，每天都在催促着成员们出题。然而每次催促完后只会带来一些“正”的出题相关的讨论，或是“负”的（比如，咕咕咕）的讨论。

这些讨论可以看作是  $n$  个离散的值  $a_1, a_2, \dots, a_n$ 。当有连续的一段  $a_i, a_{i+1}, \dots, a_{j-1}, a_j$  的和超过  $m$ ，就可以利用这些讨论来组合成一道新题目。不过为了题目的质量，同一个讨论不应当在两个或两个以上的题目中被应用；或者说选取的每一段不应当有重叠、重复使用的  $a_i$ 。注意  $m$  也可能是负的，如果 L 头子希望题目能有趣一点。

L 头子现在非常恐慌，希望你来帮他计算一下最多能出几道题。

### Input

第一行为数据组数  $t$ ，满足  $1 \leq t \leq 6$ 。

接下来共  $t$  行，每行五个整数  $n, m, u, l, r$ ，满足  $1 \leq n \leq 10^7, 0 \leq u < 2^{31}, 0 \leq |m|, |l|, |r| \leq 10^9, l \leq r$ 。

每行中的  $u, l, r$  为随机数的参数，随机数的具体算法及代码在 Note 中给出。

题目中的  $a_i$  为第  $i$  次调用以  $u, l, r$  为初始值的随机数生成器所返回的值。

### Output

对于每组数据，输出一行一个整数，最多能出几道题。

### Example

standard input	standard output
4	3
5 6 391 -20 20	1
20 19 391 -20 20	10
20 -3 391 -20 20	5
20 500000000 391096716 -1000000000	
1000000000	

### Note

随机数的生成为对变量  $u$  进行若干次位运算，并模  $(r-l+1)$  后加  $l$ 。各种语言的代码如下，对于 C/C++，Java，Python2/Python3，相应的随机数生成函数为 `RNGNextInt(RNG *)`，`RNG.nextInt()`，`RNG()`。

C/C++

```
1  #include <stdio.h>
2
3  struct RNG {
4      int u, l, r;
5  };
6
7  int RNGNextInt(struct RNG *r) {
8      r->u ^= (r->u >> 3) & 0x7fffffff;
9      r->u ^= (r->u << 5) & 0x7fffffff;
10     r->u ^= (r->u >> 1) & 0x7fffffff;
11     return (r->u % (r->r - r->l + 1)) + r->l;
12 }
13
14 int main(void) {
15     int i;
16     struct RNG rng = {.u = 391, .l = -20, .r = 20};
17     for(i = 0; i < 20; ++i)
18         printf("%d%c", RNGNextInt(&rng), " \n"[i == 19]);
19     struct RNG rngB = {.u = 391096716, .l = -(int)1e9, .r = (int)1e9};
20     for(i = 0; i < 20; ++i)
21         printf("%d%c", RNGNextInt(&rngB), " \n"[i == 19]);
22 }
```

## Java

```
1  class RNG {
2      int u, l, r;
3      public RNG(int u, int l, int r) {
4          this.u = u & 0x7fffffff;
5          this.l = l;
6          this.r = r;
7      }
8      public int nextInt() {
9          u ^= (u >> 3) & 0x7fffffff;
10         u ^= (u << 5) & 0x7fffffff;
11         u ^= (u >> 1) & 0x7fffffff;
12         return (u % (r - l + 1)) + l;
13     }
14 }
15
16 public class Main {
```

```

17 public static void main(String[] argv) {
18     RNG rng = new RNG(391, -20, 20);
19     for(int i = 0; i < 20; ++i) {
20         System.out.printf("%d%c", rng.nextInt(), i < 19 ? ' ' : '\n');
21     }
22     rng = new RNG(391096716, -1000000000, 1000000000);
23     for(int i = 0; i < 20; ++i) {
24         System.out.printf("%d%c", rng.nextInt(), i < 19 ? ' ' : '\n');
25     }
26 }
27 }

```

### Python2/Python3

```

1 from __future__ import print_function
2
3 class RNG:
4     def __init__(self, u, l, r):
5         self.u = u & 0x7fffffff
6         self.l, self.r = l, r
7
8     def __call__(self):
9         self.u ^= (self.u >> 3) & 0x7fffffff
10        self.u ^= (self.u << 5) & 0x7fffffff
11        self.u ^= (self.u >> 1) & 0x7fffffff
12        return (self.u % (self.r - self.l + 1)) + self.l
13
14 if __name__ == '__main__':
15     rng = RNG(391, -20, 20)
16     print(*(rng() for _ in range(20)))
17     rng = RNG(391096716, -10 ** 9, 10 ** 9)
18     print(*(rng() for _ in range(20)))

```

不同语言的随机数生成器所产生的结果应相同，为了便于理解，这里同时在标准输出给出了两组随机数生成器前 20 次调用的结果，标准输出内容应该为：

```

1 16 -13 14 -20 9 -19 10 0 19 -6 4 -13 -2 -3 -12 -11 -15 13 -8 4
2 -163552109 847440081 -663284450 721288435 -150123445 55934115 ↵
   -688294180 -797610444 -458986549 -149349973 14846993 736168938 ↵
   -639165076 234173553 -906965137 -659992431 760588146 949392562 ↵
   -19240170 -606974546

```



第一组数据中，对应题目的  $a_1, a_2, \dots, a_n$  为 16, -13, 14, -20, 9；第二、三组中， $a_1, a_2, \dots, a_n$  为 16, -13, 14, -20, 9, -19, 10, 0, 19, -6, 4, -13, -2, -3, -12, -11, -15, 13, -8, 4；第四组中， $a_1, a_2, \dots, a_n$  为 -163552109, 847440081, -663284450, 721288435, -150123445, 55934115, -688294180, -797610444, -458986549, -149349973, 14846993, 736168938, -639165076, 234173553, -906965137, -659992431, 760588146, 949392562, -19240170, -606974546。

## Problem D. 区间 mex

Input file: *standard input*

Output file: *standard output*

Time limit: 1 second

Memory limit: 256 mebibytes

给出一个长度为  $n$  的整数序列  $a_1, a_2, \dots, a_n$ 。再给出  $q$  组询问，每次询问给出一个整数区间  $[L, R]$ ，要求在  $[L, R]$  中找出一个整数子区间  $[l, r]$ ，满足  $L \leq l \leq r \leq R$ ，并且  $\frac{\text{len}(l, r) + 1 - \text{mex}(l, r)}{\text{len}(l, r) + 1 + \text{mex}(l, r)}$  最小。

其中  $\text{len}(l, r) = r - l + 1$ ，而  $\text{mex}(l, r)$  表示最小的不在  $a_l, \dots, a_r$  中出现的非负整数。

例如当  $[a_1, a_2, a_3, a_4] = [0, 1, 2, 5]$  时  $\text{mex}(1, 4) = 3$ ，而当  $a_5 = 100$  时， $\text{mex}(5, 5) = 0$ 。

### Input

第一行包含两个整数  $n$  和  $q$ ，含义见题目描述，满足  $1 \leq n, q \leq 10^5$ 。

第二行包含  $n$  个整数，表示序列  $a_1, a_2, \dots, a_n$ ，满足  $0 \leq a_i \leq n$ 。

接下来  $q$  行，每行包含两个整数  $L$  和  $R$ ，表示一组询问，满足  $1 \leq L \leq R \leq n$ 。

### Output

对于每组询问，输出一行，包含一个最简分数，表示该组询问的  $\frac{\text{len}(l, r) + 1 - \text{mex}(l, r)}{\text{len}(l, r) + 1 + \text{mex}(l, r)}$  最小值。即使最简分数的分母为 1，分母同样要输出，不可省略。

### Example

standard input	standard output
5 2	1/3
0 1 2 3 4	1/7
1 1	
1 3	

### Note

对于第一组询问  $[1, 1]$ ：

- 子区间  $[1, 1]$  满足  $\text{len}(1, 1) = 1$ ， $\text{mex}(1, 1) = 1$ ，对应分数为  $\frac{1+1-1}{1+1+1} = \frac{1}{3}$ ；
- 最小值为  $\frac{1}{3}$ 。

对于第二组询问  $[1, 3]$ ：

- 子区间  $[1, 1]$  对应分数为  $\frac{1}{3}$ ；
- 子区间  $[1, 2]$  对应分数为  $\frac{1}{5}$ ；
- 子区间  $[1, 3]$  对应分数为  $\frac{1}{7}$ ；
- 子区间  $[2, 2]$  对应分数为 1；
- 子区间  $[2, 3]$  对应分数为 1；
- 子区间  $[3, 3]$  对应分数为 1；
- 最小值为  $\frac{1}{7}$ 。

## Problem E. 哪个的 Scramble String

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 128 mebibytes

哪个最近在找工作，于是闲来无事刷着 LeetCode。某一天，他遇到了一道有趣的题目，叫做 Scramble String。

题目大意是说：给定两个等长的字符串，判断后者是否是前者的 Scramble String。

在这道题中，我们认为一个字符串  $y$  是另一个字符串  $x$  的 Scramble String，当且仅当可以通过如下过程将  $x$  变换为  $y$ ：

1. 选取一棵具有  $(2|x| - 1)$  个节点的满二叉树  $T$ ，为其每个叶子标上字母，使得以从上到下，从左到右的顺序访问叶子时，其上字母构成字符串  $x$ 。其中
  - $|x|$  表示字符串  $x$  的长度；
  - 满二叉树指一棵二叉树的任意节点要么没有儿子，要么具有两个儿子。
2. 执行以下操作若干次：任选  $T$  中一个节点  $n$ ，如果其有儿子，交换两个儿子。
3. 再次以从上到下，从左到右的顺序访问  $T$  的叶子时，其上字母构成字符串  $y$ 。

哪个想了很久，发现不太会做。但是在某天晚上吃着麦当劳金拱门桶的时候，哪个突然灵机一动，对于字母都不相同的情形有了一个很妙的解法，可惜，金拱门桶地方太小写不下。

现在哪个马上就要去笔试了，但他忘记怎么做这道题了，你能帮助哪个重新想出这个解法吗？

为了方便起见，我们假设字符集大小为  $n$ ，并且字符串  $x$  为  $12 \cdots n$ 。给出字符串  $y$ ，请判断  $y$  是否是  $x$  的 Scramble String。

### Input

第一行包含一个正整数  $T$ ，表示测试数据的组数。

接下来描述每组测试数据，对于每组测试数据：

第一行包含一个正整数  $n$  ( $1 \leq n \leq 5 \times 10^6$ )，表示字符集大小。

第二行包含  $n$  个整数，其中第  $i$  个数字为  $c_i$  ( $1 \leq c_i \leq n$ )，表示字符串  $y$  中第  $i$  个字母的取值。

保证每组测试数据的所有  $c_i$  两两不同，所有测试数据的  $n$  之和不超过  $10^7$ 。

### Output

对于每组数据，输出一行，如果  $y$  是  $x$  的 Scramble String，输出“**Yes**”，否则输出“**No**”（不含引号）。

### Example

standard input	standard output
2	Yes
5	No
2 1 3 4 5	
5	
3 1 5 2 4	

## Problem F. 构造题

Input file: *standard input*

Output file: *standard output*

Time limit: 1 second

Memory limit: 256 mebibytes

给定正整数  $n$ ，请你在平面上构造  $n$  个不同的点，使得任意两点的垂直平分线都穿过至少一个点。

### Input

第一行包含一个正整数  $T$  ( $1 \leq T \leq 20$ )，表示有  $T$  组测试数据。

接下来依次给出每组测试数据。对于每组测试数据：

仅一行，包含一个整数  $n$  ( $2 \leq n \leq 300$ )。

### Output

对于每组数据：

- 如果无解，输出一行 “impossible”（不含引号）；
- 否则输出  $n$  行，每一行包含两个实数  $x$  和  $y$ （用空格隔开），表示一个点  $(x, y)$ ，你的输出需要满足  $0 \leq |x|, |y| \leq 10^5$ 。

由于输出存在精度误差，因此对于你给出的解中任意两点  $(x_1, y_1)$  和  $(x_2, y_2)$ ，它们都需要满足  $|x_1 - x_2| + |y_1 - y_2| \geq 10^{-6}$ 。此外，当一个点与一条垂直平分线的距离不超过  $10^{-9}$  时，我们认为这个点落在这条直线上。

### Example

standard input	standard output
1	0 0
3	1 0
	0.5 0.8660254037844386

### Note

样例输出的三个点恰好位于一个等边三角形的顶点处。

## Problem G. 进化吧亚古兽!!!

Input file: *standard input*

Output file: *standard output*

Time limit: 4 seconds

Memory limit: 256 mebibytes

“亚古兽，进化 —— 暴龙兽！”

随着熟悉的 **BGM** 再一次响起  
被选召孩子们的冒险越来越深入  
你准备好了吗！

击败了统治法易路岛的恶魔兽之后，被选召的孩子们划着友谊的小船来到了新的神秘大陆 —— 沙拔大陆。刚登陆便遭遇了掌控大陆外围的邪恶数码兽 —— 猩猩兽，全体数码兽进化为成熟期发起合力攻击却远不敌猩猩兽，机智的太一丢出了在海边摘取的香蕉吸引了猩猩兽的注意力，众人得以逃脱。

我们的主角太一对这次失败的战斗一直耿耿于怀，便向玄内老人询问亚古兽进一步进化的方法！玄内老人如是说：“亚古兽进化的关键就是勇气徽章！咳咳…幸好在这片大陆上有一个魔法树，上面长了  $n$  片被折断的勇气徽章，我可以用魔法把你和亚古兽传送到两个节点上去，你们需要摘下相应的徽章，注意，一旦你们离开了我传送的节点，魔法树便会感应到入侵者的存在，其余节点的徽章将被隐匿。所以你们要尽快的汇合到同一个节点看看手中的两块徽章能否合成真正的勇气徽章，如果合成失败，这两块徽章就会自动回到原来的位置。接着我会使用魔法把你们传入数据空间，等魔法树解除防卫重新释放出所有的勇气徽章之后，就把你们传送到下一对节点上！赶快行动起来吧！……噢，对了，树上一共有  $(n - 1)$  条边，每条边的长度为  $w_i$ ，每片徽章有他自己的尺寸大小  $v_i$ ，任意两片徽章大小互不相同，我们每天把尺寸为  $d_i$  整数倍的每对徽章都不重复地试着合一遍，希望可以赶在被猩猩兽发现之前找到真正的勇气徽章……”

太一听后非常兴奋，转身拜托你来帮他算出每天他和亚古兽需要走的路程总长，他好提前准备食物和水~ 由于路程之和可能比较大，太一想知道路程之和  $\text{mod}(10^9 + 7)$  的值！

### Input

第一行包含一个正整数  $T$  ( $1 \leq T \leq 4$ )，表示数据组数。接下来是每组数据。

每组数据第一行包含一个整数  $n$  ( $1 \leq n \leq 10^5$ )，表示树上的徽章数。

第二行包含  $n$  个整数，第  $i$  个整数表示第  $i$  个徽章的尺寸  $v_i$  ( $1 \leq v_i \leq n$ )，各个徽章尺寸互不相同。

在接下来的  $(n - 1)$  行中，第  $i$  行包含三个整数  $x_i, y_i, w_i$  ( $1 \leq x_i, y_i \leq n, 1 \leq w_i \leq 1.1 \times 10^9$ )，代表  $x_i$  号徽章和  $y_i$  号徽章之间一条长度为  $w_i$  的边。

接下来一行包含一个整数  $m$  ( $1 \leq m \leq 10^5$ )，表示尝试的天数。

在接下来的  $m$  行中，第  $i$  行包含一个整数  $d_i$  ( $1 \leq d_i \leq n$ )，表示第  $i$  天太一将把所有  $d_i$  整数倍的每对徽章都不重复地试着合成一遍。

### Output

对于每一天，输出一行，包含一个整数，表示太一和亚古兽要走的路程之和  $\text{mod}(10^9 + 7)$  的值。

## Example

standard input	standard output
1	10
2	0
1 2	
1 2 10	
2	
1	
2	

## Note

第一天，太一要尝试将节点 1 和节点 2 上的徽章合并，所以需要经过一次 1 – 2 这条边；第二天，由于尺寸为 2 的倍数徽章只有 1 块，故不尝试任何合并方案。

## Problem H. Ange 的线性代数题

Input file: *standard input*

Output file: *standard output*

Time limit: 3 seconds

Memory limit: 256 mebibytes

定义  $n$  个数  $a_1, a_2, \dots, a_n$  的最小公倍数为  $\text{lcm}\{a_1, a_2, \dots, a_n\}$ 。

对于所有元素的最小公倍数为  $n$  的集合，定义  $H(n)$  为满足该条件的集合个数。

对于给定的正整数  $n$ ，试求  $H(\text{lcm}\{1, 2, \dots, n\}) \bmod (10^9 + 7)$ 。

### Input

仅一行，包含一个正整数  $n$ ，满足  $1 \leq n \leq 10^4$ 。

### Output

输出一行，包含一个整数，表示  $H(\text{lcm}\{1, 2, \dots, n\})$  在模  $(10^9 + 7)$  意义下的值。

### Example

standard input	standard output
3	10

### Note

$\text{lcm}\{1, 2, 3\} = 6$ ，共有  $\{2, 3\}$ ,  $\{1, 2, 3\}$ ,  $\{6\}$ ,  $\{1, 6\}$ ,  $\{2, 6\}$ ,  $\{1, 2, 6\}$ ,  $\{3, 6\}$ ,  $\{1, 3, 6\}$ ,  $\{2, 3, 6\}$ ,  $\{1, 2, 3, 6\}$  这十个集合满足条件，即  $H(6) = 10$ 。



## Problem I. 击鼓传花

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 128 mebibytes

老板最近沉迷于击鼓传花这个游戏，每天早上到公司后便会从办公室中选取  $n$  位幸运儿排成一圈顺时针编号  $1, 2, \dots, n$  并要求参与今天的游戏：游戏开始后，老板会闭着眼睛抛出手中的绣球，员工  $x$  接到绣球后开始按每 1 秒传一次的速度顺时针传球，老板每隔  $t$  秒便会将此时持球幸运儿的小名儿记在小本本上，重复  $\infty$  秒后老板终于停止了这轮游戏，而记在小本本上的幸运儿们则需要承包办公室今日的午餐外卖。贫穷的你是老板身边的红人，于是自然而然的成为老板唯一指定必须参与人员且**每天都是1号幸运儿**。

你望着干瘪的钱包想着还要存钱开龙狙，于是便打算逃离这些游戏。每天在老板上班前你都可以预见到老板今天将会选取  $n_i$  位幸运儿、首先把绣球抛到  $x_i$  号幸运儿手中、隔  $t_i$  秒记录一次名字。若今天预测到要被抽中请客，你将会在游戏开始前跑去给老板擦车溜之大吉~

### Input

第一行包含一个正整数  $m$  ( $1 \leq m \leq 10^6$ )，代表老板一共要玩  $m$  天游戏。

接下来  $m$  行，第  $i$  行包含三个整数  $n_i, x_i$  和  $t_i$ ，代表第  $i$  天你预测到的数据，满足  $1 \leq n \leq 10^6$ ,  $1 \leq x_i, t_i \leq n$ 。

### Output

对于每一天，输出一行。若你预测到在这一天你要请客吃饭，则输出“YES”，否则输出“NO”（均不含引号）。

### Example

standard input	standard output
3	YES
1 1 1	NO
2 2 2	YES
2 1 2	