



# 2018 “字节跳动杯”中国大学生程序设计竞赛 女生专场

2018 CCPC-WFINAL



2018 年 5 月 27 日

## Problem 1001. CCPC 直播

Input file:            stdin  
Output file:           stdout  
Time limit:            1 seconds  
Memory limit:         512 megabytes

在比特镇举行的中国大学生程序设计竞赛 CCPC(China Collegiate Programming Contest) 开始啦！这次比赛中加入了现场视频直播，而在直播屏幕的左下角，会显示评测队列。

比特镇的科技水平并不发达，直播分辨率并不高。准确地说，每个评测记录将被显示在 1 行 38 列的像素格上。一条评测记录由 4 个部分组成，从左往右依次为排名 (3 像素), 队名 (16 像素), 题号 (4 像素), 评测情况 (12 像素)。相邻两个部分之间由 1 像素的分隔线“|”分开。其中，排名右对齐显示，队名左对齐显示，长度不足时用空格补齐。题号一定是 4 位正整数，因此恰好占据 4 像素。评测情况则比较复杂，它由两侧的括号“[]”以及中间 10 像素组成。

```
19|qqqqq_University|1001|[XXX      ]
```

上面的例子中表示排名第 19 的队伍“qqqqq\_University”提交了题目 1001，已经通过了 30% 的测试点，故左对齐显示 3 个“X”。

```
125|quailty_U_2      |1002|[      WA      ]
```

上面的例子中表示排名第 125 的队伍“quailty\_U\_2”提交了题目 1002，评测结果为“WA”，故居中显示“WA”，左边留 4 像素空格。

```
4|quailty_U_3        |1003|[      TLE      ]
```

上面的例子中需要居中显示“TLE”，左边同样留 4 像素空格。

```
1|quailty_U_4        |1003|[      AC*      ]
```

上面的例子中这支队伍是全场第一个通过 1003 的，故在“AC”后加上字符“\*”来特殊标注。

请写一个程序，对于每条评测记录产生直播信息。

### Input

第一行包含一个正整数  $T$  ( $1 \leq T \leq 1000$ )，表示评测记录的数量。

接下来  $T$  行，每行首先是一个正整数  $rank$  ( $1 \leq rank \leq 400$ )，表示队伍的排名。

接下来一个长度不超过 16 的字符串  $S$ ，表示队名， $S$  仅由大小写字母、数字以及下划线“\_”组成。

接下来一个正整数  $prob$  ( $1001 \leq prob \leq 1013$ )，表示题号。

接下来一个字符串  $T$  ( $T \in \{\text{“Running”}, \text{“AC”}, \text{“WA”}, \text{“TLE”}, \text{“MLE”}, \text{“RTE”}, \text{“CE”}, \text{“OLE”}, \text{“PE”}, \text{“FB”}\}$ )，表示评测状态，除“Running”外均表示评测结束。若为“Running”，则还会输入一个正整数  $p$  ( $1 \leq p \leq 9$ )，表示已经通过了  $p \times 10\%$  的测试点。若为“FB”，则表示全场第一个通过该题，应显示“AC\*”。

## Output

对于每条评测记录，按要求输出一行一个长度为 38 的字符串，即直播显示效果。

## Examples

stdin	stdout
5	19 qqqqq_University 1001 [XXX ]
19 qqqqq_University 1001 Running 3	125 quailty_U_2  1002 [ WA ]
125 quailty_U_2 1002 WA	4 quailty_U_3  1003 [ TLE ]
4 quailty_U_3 1003 TLE	1 quailty_U_4  1003 [ AC* ]
1 quailty_U_4 1003 FB	2 qqqqq  1001 [ AC ]
2 qqqqq 1001 AC	

## Problem 1002. 口算训练

Input file: `stdin`  
Output file: `stdout`  
Time limit: 4 seconds  
Memory limit: 512 megabytes

小 Q 非常喜欢数学，但是他的口算能力非常弱。因此他找到了小 T，给了小 T 一个长度为  $n$  的正整数序列  $a_1, a_2, \dots, a_n$ ，要求小 T 抛出  $m$  个问题以训练他的口算能力。

每个问题给出三个正整数  $l, r, d$ ，小 Q 需要通过口算快速判断  $a_l \times a_{l+1} \times \dots \times a_{r-1} \times a_r$  是不是  $d$  的倍数。

小 Q 迅速地回答了出来，但是小 T 并不知道正确答案是什么，请写一个程序帮助小 T 计算这些问题的正确答案。

### Input

第一行包含一个正整数  $T(1 \leq T \leq 10)$ ，表示测试数据的组数。

每组数据第一行包含两个正整数  $n, m(1 \leq n, m \leq 100000)$ ，分别表示序列长度以及问题个数。

第二行包含  $n$  个正整数  $a_1, a_2, \dots, a_n(1 \leq a_i \leq 100000)$ ，表示序列中的每个数。

接下来  $m$  行，每行三个正整数  $l, r, d(1 \leq l \leq r \leq n, 1 \leq d \leq 100000)$ ，表示每个问题。

### Output

对于每个问题输出一行，若是倍数，输出“**Yes**”，否则输出“**No**”。

### Examples

stdin	stdout
1	Yes
5 4	No
6 4 7 2 5	No
1 2 24	Yes
1 3 18	
2 5 17	
3 5 35	

## Problem 1003. 缺失的数据范围

Input file: `stdin`  
Output file: `stdout`  
Time limit: 1 seconds  
Memory limit: 512 megabytes

著名出题人小 Q 出过非常多的题目，在这个漫长的过程中他发现，确定题目的数据范围是非常痛苦的一件事。

每当思考完一道题目的时间效率，小 Q 就需要结合时限以及评测机配置来设置合理的数据范围。

因为确定数据范围是一件痛苦的事，小 Q 出了非常多的题目之后，都没有它们设置数据范围。对于一道题目，小 Q 会告诉你他的算法的时间复杂度为  $O(n^a \log^b n)$ ，且蕴含在这个大  $O$  记号下的常数为 1。同时，小 Q 还会告诉你评测机在规定时限内可以执行  $k$  条指令。小 Q 认为只要  $n^a (\lceil \log_2 n \rceil)^b$  不超过  $k$ ，那么就是合理的数据范围。其中， $\lceil x \rceil$  表示最小的不小于  $x$  的正整数，即  $x$  上取整。

自然，小 Q 希望题目的数据范围  $n$  越大越好，他希望你写一个程序帮助他设置最大的数据范围。

### Input

第一行包含一个正整数  $T$  ( $1 \leq T \leq 1000$ )，表示测试数据的组数。

每组数据包含一行三个正整数  $a, b, k$  ( $1 \leq a, b \leq 10, 10^6 \leq k \leq 10^{18}$ )，分别描述时间复杂度以及允许的指令数。

### Output

对于每组数据，输出一行一个正整数  $n$ ，即最大可能的  $n$ 。

### Examples

stdin	stdout
3	4347826
1 1 100000000	2886
2 1 100000000	48828
1 3 200000000	

## Problem 1004. 寻宝游戏

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 512 megabytes

小 Q 最近迷上了一款寻宝游戏，这款游戏中每局都会生成一个  $n \times m$  的网格地图，从上往下依次编号为第 1 行到第  $n$  行，从左往右依次编号为第 1 列到第  $m$  列。每个格子上都有不同数量的金币，第  $i$  行第  $j$  列的格子上的金币数量为  $a_{i,j}$ 。

小 Q 一开始位于  $(1, 1)$ ，每次他可以往右或者往下走，每当他经过某个格子时，他就可以拿走这个格子上的所有金币。小 Q 不能走出这个地图，当小 Q 不能再行动时，游戏结束。显然当且仅当小 Q 位于  $(n, m)$  时，游戏才会结束。

一轮游戏的得分为这一轮中收集到的金币总量，而在游戏开始前，因为小 Q 是超级 VIP 用户，所以他有  $k$  次机会交换某两个格子中的金币数。这  $k$  次机会不一定要用完，请写一个程序帮助小 Q 在一轮内拿到尽可能多的金币。

### Input

第一行包含一个正整数  $T$  ( $1 \leq T \leq 10$ )，表示测试数据的组数。

每组数据第一行包含三个整数  $n, m, k$  ( $2 \leq n, m \leq 50, 0 \leq k \leq 20$ )，分别表示地图的长宽以及交换的次数。

接下来  $n$  行，每行  $m$  个整数  $a_{i,j}$  ( $0 \leq a_{i,j} \leq 10^6$ )，依次表示每个格子中金币的数量。

### Output

对于每组数据，输出一行一个整数，即能收集到的金币数量的最大可能值。

### Examples

stdin	stdout
2	34
3 4 0	81
1 2 3 4	
9 8 7 6	
5 4 7 2	
5 5 1	
9 9 9 0 0	
0 0 9 0 0	
0 0 0 0 0	
0 0 9 0 0	
9 0 9 9 9	

## Problem 1005. 奢侈的旅行

Input file: `stdin`  
Output file: `stdout`  
Time limit: 7 seconds  
Memory limit: 512 megabytes

高玩小 Q 不仅喜欢玩寻宝游戏, 还喜欢一款升级养成类游戏。在这个游戏的世界地图中一共有  $n$  个城镇, 编号依次为 1 到  $n$ 。

这些城镇之间有  $m$  条单向道路, 第  $i$  条单向道路包含四个参数  $u_i, v_i, a_i, b_i$ , 表示一条从  $u_i$  号城镇出发, 在  $v_i$  号城镇结束的单向道路, 因为是单向道路, 这不意味着小 Q 可以从  $v_i$  沿着该道路走到  $u_i$ 。小 Q 的初始等级  $level$  为 1, 每当试图经过一条道路时, 需要支付  $cost = \log_2 \frac{level + a_i}{level}$  点积分, 并且经过该道路后, 小 Q 的等级会提升  $a_i$  级, 到达  $level + a_i$  级。但是每条道路都会在一定意义上歧视低消费玩家, 准确地说, 如果该次所需积分  $cost < b_i$ , 那么小 Q 不能经过该次道路, 也不能提升相应的等级。

**注意:** 本游戏中等级为正整数, 但是积分可以是任意实数。

小 Q 位于 1 号城镇, 等级为 1, 现在为了做任务要到  $n$  号城镇去。这将会是一次奢侈的旅行, 请写一个程序帮助小 Q 找到需要支付的总积分最少的一条路线, 或判断这是不可能的。

### Input

第一行包含一个正整数  $T (1 \leq T \leq 30)$ , 表示测试数据的组数。

每组数据第一行包含两个整数  $n, m (2 \leq n \leq 100000, 1 \leq m \leq 200000)$ , 表示城镇数和道路数。

接下来  $m$  行, 每行四个整数  $u_i, v_i, a_i, b_i (1 \leq u_i, v_i \leq n, u_i \neq v_i, 0 \leq a_i \leq 10^9, 0 \leq b_i \leq 60)$ , 分别表示每条单向道路。

### Output

对于每组数据, 输出一行一个整数, 即最少所需的总积分的整数部分, 如: “4.9999”输出“4”, “1.0”输出“1”。若不存在合法路线请输出“-1”。

### Examples

stdin	stdout
1 3 3 1 2 3 2 2 3 1 6 1 3 5 0	2

## Problem 1006. 对称数

Input file: `stdin`  
Output file: `stdout`  
Time limit: 15 seconds  
Memory limit: 512 megabytes

小 Q 认为, 偶数具有对称美, 而奇数则没有。

给定一棵  $n$  个点的树, 任意两点之间有且仅有一条直接或间接路径。这些点编号依次为 1 到  $n$ , 其中编号为  $i$  的点上有一个正整数  $a_i$ 。

定义集合  $S(u, v)$  为  $u$  点到  $v$  点的唯一最短路径上经过的所有点  $x$  (包括  $u$  和  $v$ ) 对应的正整数  $a_x$  的集合。小 Q 将在  $m$  个  $S(u, v)$  中寻找最小的对称数。因为偶数具有对称美, 所以对称数是指那些出现了偶数次 (包括 0 次) 的正整数。

请写一个程序, 帮助小 Q 找到最小的对称数。

### Input

第一行包含一个正整数  $T (1 \leq T \leq 10)$ , 表示测试数据的组数。

每组数据第一行包含两个正整数  $n, m (1 \leq n, m \leq 200000)$ , 分别表示点数和询问数。

第二行包含  $n$  个正整数  $a_1, a_2, \dots, a_n (1 \leq a_i \leq 200000)$ , 依次表示每个点上的数字。

接下来  $n - 1$  行, 每行两个正整数  $u_i, v_i (1 \leq u_i, v_i \leq n, u_i \neq v_i)$ , 表示一条连接  $u_i$  和  $v_i$  的双向树边。

接下来  $m$  行, 每行两个正整数  $u_i, v_i (1 \leq u_i, v_i \leq n)$ , 依次表示每个询问。

### Output

对于每个询问输出一行一个正整数, 即最小的对称数。

### Examples

stdin	stdout
1	2
5 3	1
1 2 2 1 3	1
1 2	
1 3	
2 4	
2 5	
2 3	
1 4	
2 5	



## Problem 1007. 赛题分析

Input file: `stdin`  
Output file: `stdout`  
Time limit: 1 seconds  
Memory limit: 512 megabytes

著名出题人小 Q 每次比赛后都会写一份《赛题分析》，包含比赛概况、每题的参考算法以及一些统计数值。

对于一道题来说，小 Q 会统计最短的验题人代码长度 (Shortest judge solution) 以及赛内参赛队伍最短的 AC 代码长度 (Shortest team solution)。

统计验题人代码长度比较容易，因为验题人最多也不会超过 20 个。但是统计选手代码长度就不容易了，因为大赛区动辄三四百支队伍。

请写一个程序，帮助小 Q 统计最短代码长度。

### Input

第一行包含一个正整数  $T(1 \leq T \leq 13)$ ，表示赛题数量。

每道题第一行包含两个整数  $n, m(2 \leq n \leq 20, 0 \leq m \leq 500)$ ，分别表示验题人数量以及 AC 了该题的队伍数量。

第二行包含  $n$  个正整数  $a_1, a_2, \dots, a_n(50 \leq a_i \leq 65536)$ ，依次表示每个验题人的代码字节数。

第三行包含  $m$  个正整数  $b_1, b_2, \dots, b_m(50 \leq b_i \leq 65536)$ ，依次表示每支 AC 队伍的代码字节数。若  $m = 0$  则该行为空行。

### Output

对于第  $i(1 \leq i \leq T)$  道题，输出三行，第一行输出“Problem  $x$ :”，其中  $x = i + 1000$ 。

第二行输出“Shortest judge solution:  $y$  bytes.”，其中  $y$  表示最短的验题人代码字节数。

第三行输出“Shortest team solution:  $z$  bytes.”，其中  $z$  表示最短的选手代码字节数，若不存在请输出“N/A”。

注意：间隔都是一个空格。

### Examples

stdin	stdout
2	Problem 1001:
3 2	Shortest judge solution: 1460 bytes.
3627 1460 5288	Shortest team solution: 2365 bytes.
2365 2671	Problem 1002:
2 0	Shortest judge solution: 5510 bytes.
5510 7682	Shortest team solution: N/A bytes.

## Problem 1008. quailty 算法

Input file: `stdin`  
Output file: `stdout`  
Time limit: 2 seconds  
Memory limit: 512 megabytes

二分图又称作二部图，是图论中的一种特殊模型。设  $G = (V, E)$  是一个无向图，如果点集合  $V$  可分割为两个互不相交的子集  $A, B$ ，并且图中的每条边  $(i, j)$  所关联的两个顶点  $i$  和  $j$  分别属于这两个不同的顶点集 ( $i \in A, j \in B$ )，则称图  $G$  为一个二分图。

一个“匹配”是指一组  $E$  的子边集  $M$ ，满足  $M$  中任意两条边都没有公共顶点。“最大匹配”是指  $M$  中边数最多的一个匹配。“最小费用最大匹配”是指在边数最多的基础上，所选择的边的权值总和最小的匹配。

小 T 最近刚刚学会了二分图最大权匹配的 KM(Kuhn-Munkres) 算法，知道 KM 算法稍加修改就能计算二分图最小费用最大匹配。小 T 兴奋地把这个算法讲给小 Q 听，而小 Q 却吹牛说他自己发明了一个 quailty 算法，可以在低于线性的时空复杂度里求解二分图最小费用最大匹配。

小 T 不相信，于是给定  $n$  和一个长度为  $n$  的正整数数组  $a_1, a_2, \dots, a_n$ ，写了一段代码根据  $a[]$  生成了一个很大的带权二分图，代码如下：

```
int ca = n;

long long cb = 0;

for(int i = 1; i <= n; i++){
    for(int j = 1; j < i; j++){
        cb++;

        long long k = cb;

        addedge(i, k, a[i] xor a[j]);

        addedge(j, k, a[i] xor a[j]);
    }
}
```

相关说明：

- $ca$  表示  $A$  中的点数， $cb$  表示  $B$  中的点数，点都是从 1 开始连续编号的。
- “`addedge(x,y,z)`”表示添加一条边权为  $z$  的无向边，两端点分别为  $A$  中的  $x$  点和  $B$  中的  $y$  点。
- “`xor`”表示按位异或的运算符  $\oplus$ 。按位异或的运算符是双目运算符，具有交换律，即  $i \oplus j = j \oplus i$ 。按位异或可以理解成被运算的数字的二进制位对应位如果相同，则结果的该位置为 0，否则为 1，例如： $1(01) \oplus 2(10) = 3(11)$ 。按位异或还可以理解成参与运算的数字的每个二进制位都进行了不进位的加法，例如： $3(11) \oplus 3(11) = 0(00)$ 。

面对如此庞大的图，小 Q 自然不能算出来，于是他向你紧急求助。请写一个程序，求出最大匹配所需的最小费用。

## Input

第一行包含一个正整数  $T(1 \leq T \leq 10)$ ，表示测试数据的组数。

每组数据第一行包含一个正整数  $n(1 \leq n \leq 300000)$ 。

第二行包含  $n$  个正整数  $a_1, a_2, \dots, a_n(1 \leq a_i \leq 10^9)$ 。

## Output

对于每组数据，输出一行一个整数，即最大匹配所需的最小费用。

## Examples

stdin	stdout
2	30
3	20
8 2 5	
5	
9 3 5 7 1	

Input file:	<code>stdin</code>
Output file:	<code>stdout</code>
Time limit:	1 seconds
Memory limit:	512 megabytes

## Problem 1010. 回文树

Input file: `stdin`  
Output file: `stdout`  
Time limit: 3 seconds  
Memory limit: 512 megabytes

给定一棵  $n$  个点的树，任意两点之间有且仅有一条直接或间接路径。这些点编号依次为 1 到  $n$ ，其中编号为  $i$  的点上有一个正整数  $a_i$ 。你可以认为每个数  $a_i$  都是在  $[1, n]$  里等概率随机挑选的。

令  $S(u, v)$  表示在树上  $u$  到  $v$  的唯一最短路径上，按照离  $u$  从近到远依次考虑每个点，将它们上面的数字  $a$  按顺序写下连成的字符串。若一个字符串正着读和倒着读相等，则称它为回文串，比如“1 2 1”和“1 3 3 1”是回文串，但是“1 21”不是回文串。

请写一个程序，统计有多少个正整数对  $(u, v)$  满足  $1 \leq u \leq v \leq n$ ，且  $S(u, v)$  是回文串。

### Input

第一行包含一个正整数  $T(1 \leq T \leq 20)$ ，表示测试数据的组数。

每组数据第一行包含一个正整数  $n(1 \leq n \leq 100000)$ ，表示点数。

第二行包含  $n$  个正整数  $a_1, a_2, \dots, a_n(1 \leq a_i \leq n)$ ，依次表示每个点上的数字。你可以认为每个数  $a_i$  都是在  $[1, n]$  里等概率随机挑选的。

接下来  $n - 1$  行，每行两个正整数  $u_i, v_i(1 \leq u_i, v_i \leq n, u_i \neq v_i)$ ，表示一条连接  $u_i$  和  $v_i$  的双向树边。

### Output

对于每组数据，输出一行一个整数，即满足条件的点对  $(u, v)$  的数量。

### Examples

stdin	stdout
2	4
3	3
1 2 1	
1 2	
2 3	
2	
2 2	
1 2	

## Problem 1011. 代码派对

Input file: `stdin`  
Output file: `stdout`  
Time limit: 3 seconds  
Memory limit: 512 megabytes

比特镇的算法竞赛选手们应邀来到了小 Q 家中参加盛大的“代码派对”。小 Q 计划在家中举行一场 ACM 赛制的友谊赛，3 个人为一队进行比赛。

为了让比赛更加有趣，也为了让大家能多多了解他人，小 Q 在花园的地上画出了一个  $1000 \times 1000$  的网格图，从上往下依次编号为第 1 行到第 1000 行，从左往右依次编号为第 1 列到第 1000 列。小 Q 让每一名选手选择一个网格图内平行坐标轴的子矩形  $(x_1, y_1) - (x_2, y_2)$ ，表示以  $(x_1, y_1)$  和  $(x_2, y_2)$  为对顶点的矩形，包含所有满足  $x \in [x_1, x_2]$  且  $y \in [y_1, y_2]$  的格子。

小 Q 规定， $i, j, k (1 \leq i < j < k \leq n)$  三人组成一支三人队伍能参赛，当且仅当这三个人选择的子矩形包含至少一个公共格子。请写一个程序，帮助小 Q 统计有多少种三人组合组成的队伍可以参赛。

### Input

第一行包含一个正整数  $T (1 \leq T \leq 10)$ ，表示测试数据的组数。

每组数据第一行包含一个正整数  $n (3 \leq n \leq 100000)$ ，表示选手人数。

接下来  $n$  行，每行四个正整数  $x_1, y_1, x_2, y_2 (1 \leq x_1 \leq x_2 \leq 1000, 1 \leq y_1 \leq y_2 \leq 1000)$ ，分别表示每个选手选择的矩形的顶点坐标。

### Output

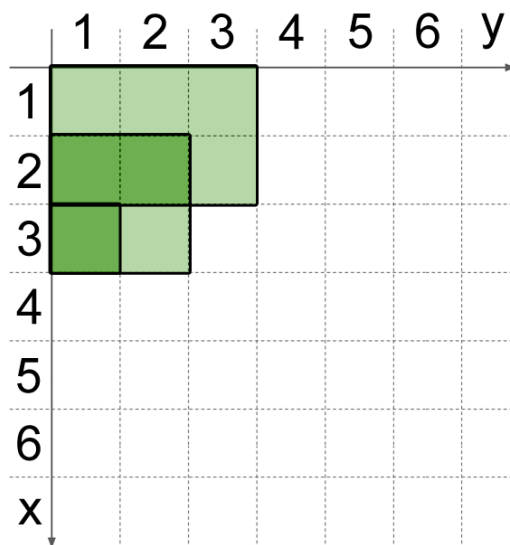
对于每组数据，输出一行一个整数，即满足条件的三人组合数量。

### Examples

stdin	stdout
2	0
3	4
3 1 3 1	
1 1 2 3	
2 1 3 2	
5	
1 1 4 5	
2 1 3 2	
2 2 3 3	
4 5 4 5	
1 2 2 4	

## Notes

第一组数据如下图:



第二组数据如下图:

