

注意事项：

- 1、使用 C++ 评测
- 2、不开 O2，没有 C++11
- 3、可以使用万能头，但是不能使用 pbrds
- 4、不要内嵌汇编
- 5、代码限制：200K
- 6、类型 `long long` \rightarrow `lld`
- 7、`__builtin` 可以使用
- 8、所有 $T \leq 20$

1. 石子合并

(stone.cpp)

(stone.in/out)

时间限制：1s/空间限制 256MB

【题目描述】

Wwq 十分喜欢写小说，今天，他正在构建他新书的世界观。

在 Wwq 的新世界中，有两种石头，为了使大家方便区分，不如称为阳极石和阴极石（当然原本的名字要中二一千零二十四倍）。每块阳极石和阴极石，都有一个指数 x 量化他所蕴含的能量， x 越大，其所蕴含的能量越大（ x 可以为负数，至于为什么可以为负数，因为 Wwq 觉得这样很帅）。在设定上，你是无法直接提取阳极石和阴极石中的能量的，你只能通过一种叫湮灭的操作来间接获得其中的能量，一块能量为 a 的阳极石和一块能量为 b 的阴极石相遇会产生湮灭反应，释放 $a-b$ 的能量（ $a-b$ 可以为负数，这意味着他需要额外消耗 $b-a$ 的能量才能进行湮灭反应，你可以参考高中化学的吸热放热反应）。

现在 Wwq 设定了 n 块阳极石和 m 块阴极石，他比较想知道他最多能得到多少的能量。（显然的，石头没有必要都用完）

【输入格式】

多组测试数据，在输入的第一行会给出测试数据组数 T 。

每组测试数据一共三行，第一行为两个整数 N, M 。

第二行一共 N 个整数，表示 N 块阳极石。

第三行一共 M 个整数，表示 M 块阴极石。

【输出格式】

每组数据输出一行。

【输入输出样例】

stone. in	stone. out
2	20
5 6	738
6 7 8 9 10	
2 3 4 5 6 7	
4 7	
123 32 213 431	
59 9034 21 4 54 123 4	

【数据规模约定】

对于 30%的数据， N, M 不超过 10^3 。

对于 100%的数据， N, M 不超过 10^5 。

输入的所有数据均保证在 `int` 范围内。

2. 校验码

(code. cpp)

(code. in/out)

时间限制：1s/空间限制 256MB

【题目描述】

Wwq 为了能破解室友的 QQ 密码特地选修了信科密码学的课。然后，他终于在开学一个月后意识到这个课好像和他想象的完全不一样。

今天老师上课介绍了校验码。校验码的作用之一是检查数据在传输过程中是否出错，其历史可以追溯到犹太人写圣经。聪明的希伯来人在誊写圣经的时候为了保证绝对正确，除了提醒自己要仔细和反复检查外，他们还采取了一种特殊的方法——他们给希伯来字母编了码，这样圣经的每一行都是一串数字，他们把这串数字经过一定的运算后得到一个数字，再转化回希伯来字母。这样圣经的单独每一行都有一个校验字母，每一列也有一个校验字母，他们只要在誊写完一页圣经后立刻对比所有的校验字母是否和原圣经相同即可。（我们一般认为在同一行犯错两次的概率比较低）

Wwq 被这种简单而有效的数学美打动了，他决定也用这种方法来生成他昨晚做题时得到的一个矩阵的校验码。（虽然我们根本不知道 Wwq 生成校验码能做什么，也许是单纯为了好玩）

为了处理 Wwq 最头疼的溢出问题，Wwq 在计算校验码时直接使用了位运算（or），这样他处理任何数都不用考虑溢出的问题了。

也就是说，Wwq 现在有一个 $N * M$ 的矩阵，矩阵中每个数都是一个 K 位 2 进制数，Wwq 会把矩阵每一行的数都或起来，得到 N 个数，他也会把每一列的数都或起来，得到 M 个数，这样他就有了 $N + M$ 个校验码。

然后——Wwq 发现他的矩阵在计算后得到的 $N + M$ 个校验码全是 $2^K - 1 \dots\dots$

Wwq 现在有了个新问题，有多少种不同的情况会导致得到的校验码全是 $2^K - 1$ 呢？

即，有多少个 $N * M$ 的矩阵满足：其所有数字都是 K 位 2 进制数，且每一行每一列的或都是 $2^K - 1$ 。

这个问题对于 Wwq 来说有点太难了，所以他找到了你，能帮帮可爱的 Wwq 吗？

啊，当然他知道方案数非常多，所以你只要求出方案数对 $10^9 + 7$ 的模即可。

【输入格式】

多组测试数据，在输入的第一行会给出测试数据组数 T 。

每组测试数据一共一行，为三个整数 N, M, K 。

【输出格式】

输出每组数据一行，即答案。

【输入输出样例】

code. in	code. out
4	1
1 1 31	1
1 10 2	7
2 2 1	625
2 3 2	

【数据规模约定】

对于 30% 的数据， N, M 不超过 10。

对于 100% 的数据， N, M 不超过 50， K 不超过 31。

3. 三足鼎立

(triangle.cpp)

(triangle.in/out)

时间限制：2s/空间限制 256MB

【题目描述】

Wwq 最近在玩一个烧脑手游——三足鼎立。在游戏中你需要管理 N 个城市，每两个城市之间都有一条无向通路（你可以认为这是一个 N 个点的完全图）。你管理城市的方式是让 M 个不同的帮派各管理一部分的道路，保证每条边都被一个帮派所管理。

我们都知道道路的管理权是十分重要的——新航路的开辟在很大程度上是因为奥斯曼土耳其帝国对陆上交通的垄断。很不幸的， M 个不同的帮派都十分仇视对方，为了国家的稳定考虑，如果对于三个城市，链接他们的三条边中有两条边属于同一个帮派，那么管理剩下一条边的帮派就会被压制，最后爆发矛盾。如果对于三个城市，链接他们的三条边都属于同一个帮派，那么他们就会形成一个小据点——这也是你不想看见的。只有三条边都属于不同的帮派，才会形成一个相对和平的稳定态——三足鼎立，这样的三角形被称为稳定三角形。

游戏的预设状态下， N 个城市间已经有 Q 条边被部分帮派管理，你可以决定剩下边由哪些帮派管理。

现在 Wwq 想要知道如果他穷举所有的分配方案，他能得到多少个不同的稳定三角形。

不同的稳定三角形的定义

——如果两个三角形他们有一个顶点不同，那么这是两个不同的三角形。

——如果两个三角形他们有一条对应边的管理帮派不同，那么这是两个不同的三角形

即，两个三角形相同当且仅当他们拥有相同的顶点，且对应的边所属的帮派也相同。

Wwq 知道答案很大，而且他知道同学们都不想写取模操作，所以你只要输出结果对 2^{32} 的模即可。

【输入格式】

多组测试数据，在输入的第一行会给出测试数据组数 T 。

每组测试数据一共 $Q+1$ 行，

第一行为三个整数 N, M, Q 。

表示有 N 个城市， M 个帮派， Q 条被预先分配了帮派的边。

城市的编号为 $1 \sim N$ ，帮派的编号为 $1 \sim M$ 。

接下来 Q 行，每行三个数字 u, v, c

表示从 u 到 v 的双向道路被编号为 c 的帮派接管。

【输出格式】

输出每组数据一行正数，即答案。

【输入输出样例】

triangle. in	triangle. out
3	10
4 6 5	504
1 2 1	76616
1 3 2	
1 4 3	
2 3 4	
2 4 5	
9 4 17	
1 7 3	
8 2 4	
2 7 3	
8 5 2	
8 7 1	
5 3 4	
7 9 2	
9 4 2	
7 6 4	
2 3 1	
4 8 2	
5 2 3	
1 9 3	
9 5 3	
6 2 1	

4 6 4	
8 3 4	
10 10 2	
1 2 3	
1 3 4	

【数据规模约定】

对于 30% 的数据， N, M 不超过 100。

对于 100% 的数据， N, M 不超过 10^5 ， Q 不超过 2×10^5 。

对于后面 70% 的数据，单个数据中 N, M 超过 100 的数据组数不超过 3。

【样例解释】

仅解释第一个样例

一共是 4 个城市，6 个帮派，5 条边

所以最后满足条件的三角形有 (1, 2, 3) (1, 2, 4) 这两个三角形的每条边所属帮派是固定的

剩下两个三角形 (1, 3, 4) (2, 3, 4) 由于 (3, 4) 这条边我们可以随意分配帮派，对于每个三角形，有 4 种不同可能：对于三角形 (1, 3, 4)，(3, 4) 边的帮派可以是 (1456)；对于三角形 (2, 3, 4)，(3, 4) 边的帮派可以是 (1236)

所以一共是 $2 + 2 * 4 = 8$

请注意读入格式，m 是帮派数，q 才是边的数量