solution

选课

这个题作为签到题应该算是刚好适合吧.
令cnt_a为选了a这门课的人数,$rep_{a,b}$为既选了a又选了b的人数.
根据我们小学二年级就学过的容斥原理,询问的答案为$cnt_a + cnt_b - rep_{a,b} \times 2$.
开大小为m的数组cnt和大小为m^2的数组rep即可轻松获得30分.
至于满分做法,会stl的都懂,只要开m个map代替大小为m^2的数组即可解决空间的问题.($multiset$会在m很小的时候复杂度爆炸,因此不能使用.)

但是本题并没有结束.这个代码只能获得83分.你没看错,这个标算只有83分.
那怎么办?考虑简单优化一下,把a, b, c从小到大排个序,这样维护rep数组的复杂度就会降为原来的一半,也就是所谓"常数减小了一半".
但是还不够,再加个快读或者使用$unordered_map$代替便可通过此题.

倘若不使用快读也可以,因为时间要求最高的数据竟然是$m \le 3000$ 的,这样用m^2数组的30分算法反而能够把每次操作的时间复杂度降为$O(1)$从而将时间复杂度控制在时限以内.
这样本题就可迎刃而解,可以去吃麻婆豆腐了.

善良的出题人

算法来源于Codeforces Round #708 (Div. 2) 的一道只有1700的题，赛后在群里大佬还给群友们进行讨论，教学。而且我的这题比unr场的这题简单

1497E1	Square-free division (easy version) data structures, dp, greedy, math, number theory, two pointers		1700	x4405

大家都补题了，所以应该很简单

补题情况（赛时好几位大佬切了这题

	* lmsh	1650			225			525	450
	* hangzai	1200		150	225	225	150		450
	* void_f	1050					150		450
	* wanglezz	600			225	225	150		
	* Pedestrian1	600					150		450
	* AngelKnows	450							450

15分算法：分解质因数用根号算法

（毕竟是第二题，你不补题怪我咯）

满分算法：log分解质因数

线性筛预处理，每次分解质因数除的都是线性筛里存的因子

线性筛https://blog.csdn.net/zzkksunboy/article/details/72860997赵章凯的博客 orz

注意

不要作死使用map、unordered_map (若使用则为15分)

hzy是一位善良的出题人，这么大的输入cin可以过

买路钱

算法一：对于25%的数据：

求最短距离用Warshall-Floyd算法，N个点的图可以用$O(N^3)$的时间求出所有点对的最短距离，对于每次询问求一遍所有点对的最短距离，时间复杂度$O(QN^3)$

算法二：对于50%的数据：

求从顶点1到其他所有顶点的最短距离用Dijkstra算法；时间复杂度$O(QMlogN)$

算法三：对于50%的数据：

对于每个询问，我们不将它的边权改为2，而是将它删除，剩下的所有边权都仍然是1，所以计算最短距离可以用BFS，时间复杂度$O(QM)$

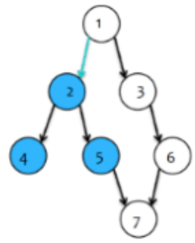
算法四：对于75%的数据：

如果将图分层，同一层中的边，对产生最短路没有影响，在初始的图上执行一次BFS，就能枚举最短路上用到的所有边，删除任意最短路上都用不到的边的询问。时间复杂度$O(Q+TM)(=O(TM))$（因为$Q<=M$），其中T是输出中出现的不同整数的数量

算法五：对于 100%的数据：

重复计算同一条边，时间复杂度太高，在前面的算法中 T 次 BFS 做了相同的事情，如图，如果删除边 1 -> 2（左上角蓝色的边）之后，所有的蓝色的点都会受到影响。

在最多 Q 次的 BFS（或者 DFS）里，检查边 u -> v 只在顶点 u 第一次变成不可到达时（路径消失时）执行一次，所以，时间复杂度是 O(M)。



算法六：对于 100%的数据：

相比较于“删除”边，“增加”边也许更容易做。反过来看所有询问，增加了最后的 i 个询问就相当于删除了前面 Q-i 个询问的边。

算法7 85~100%

对询问反过来做，Q遍spfa。在老年机上用pascal能拿85分只T掉3个点。所以OJ上分数应该更高,也许能切了这道题。

所以题目中我说这道题数据弱，暴力可以艹过去的 。妥妥的良心送分题。

cake

注意到面积最大的小正方形面积是行最大值乘列最大值。而行与列互相不冲突，因此可以将行与列分开考虑，每次操作完输出行的最大值与列的最大值的乘积。

使用 set/mulitset 维护切痕的位置以及切痕左/右区间的长度即可。时间复杂度 $\mathcal{O}(klog(n+m))$ ，期望得分50。

注意到可以将整个操作按照时间顺序逆序进行，即：初始时有一些区间，每次合并两个相邻的区间，求每次合并后最长区间的长度。

而每次合并两个区间仅会对当前划痕与相邻两个划痕产生影响，因此可以通过链表维护每一个划痕在空间上的前驱与后继，这样，每次合并两个区间只需要 $\mathcal{O}(\text{常数})$ 。同时，由于合并两个区间只会产生一个更大的区间，因此每次合并后，区间长度的最大值只会增大，不会减小。这样，每次合并两个区间，就只需要记录一下初始所有区间的最长长度 $\max l$，每次合并后与新区间长度取一次 $\max$ ，最后倒序输出即可。因此计算答案的时间复杂度为 $\mathcal{O}(k)$。

由于划痕产生是按照时间顺序输入的，我们并不能直接将其按照空间顺序链接成为链表，否则在最坏情况时间复杂度会退化为$\mathcal{O}\left(k^2\right)$，因此必须先对划痕进行排序。同时，这里必须使用桶排，才能使排序时间复杂度控制在 $\mathcal{O}(n+m)$。

最终时间复杂度$\mathcal{O}(k+n+m)$，期望得分100。

半精灵数

关于本题的故事

本题背景作品:《世界树的游戏》《神临麻雀》
世界树是个神教,看了之后每天满脑子赞美女神,大家千万不要去看.
神临麻雀是我写的小说,讲述了主角一行人被神降临之后带领麻雀竞赛没落的绿源中学一路披荆斩棘,以凡人之躯对抗神明的故事.
说远了,回到题解.

题解

首先说16分算法.
预处理$pre[p][i]$为$1 \to i$之间p的半精灵数的个数,复杂度为9×10^6.
然后利用前缀和思想$O(1)$即可回答每一个询问.

再说第2个16分算法,$p=1$,即为求不存在1的数字的个数.
利用补集思想,容斥原理以及小学奥数的基础,求出$1 \to i$中存在1的数字的个数不是问题,
然后利用前缀和思想一减即可求得答案.

对于第3个12分部分分,既然提交答案,那么就是八仙过海各显神通,答案如下:

```
150094635296999120
483322818300333431
616635121567666958
666614091501667155
683259728102334017
727711182568778395
749932403359286342
758259728102334017
772148616991222904
```

结合前两个部分分可以获得48分的好成绩.

但是本题其实是一个非常简单的数位dp,大家不要被题目吓到了,p只有一位,只要直接把板子套上去就可以了.
令$dp[now][p][rem][0/1]$表示当前枚举到第now位,询问的p,当前数字模$p=rem$,当前的数字是否已经出现过p的数字总数.
用dfs进行dp会更方便.
利用前缀和思想,轻松得到答案.
判断结束条件的时候就是`have&&rem||!have&&!rem`,含有p且余数不为0或者不含有p且余数为0时生效,和题目中说的一样,出题人写标程的时候脑子瓦特搞反了.
显然时间复杂度为均摊的$O(lg^2n)$,小小100000组数据不足道也.
至此直接提交AC走人,回去看女神了.

mess

考虑枚举出每个数字所有可能转移得到的值。对这些值取交集，得到的集合中出现 n 次的即为可能的最终状态。再一一计算到达最终状态的步数取最小值即可。

显然，计算一个数字可以到达哪个状态的时候也可以同时计算出该数字到达该状态需要多少步数。因此统计每个数字对应的结果与计算到达当前状态需要多少步数可以同时进行。

由于每个数字先除后乘与先乘后除得到的结果可能不同，例如 $3/2*2=2, 3*2/2=3$，可以推算出每个数字对应的结果数与其二进制表示中1的个数有关，在最坏情况，即其二进制表示为 $(1111...1)_2$ 时，对应的状态总数为 log^2n，即统计每个数字对应的结果的最坏时间复杂度为 $\mathcal{O}\left(nlog^2n\right)$。

注意到当一个数字二进制最低位为0时，对它进行先乘后除与先除后乘得到的结果是一样的，因此，为了避免对同一结果的重复计算，我们只需要对二进制最低位为0的状态进行先除后乘的处理。

最后扫描一次全部状态，对所以可能的最终状态需要的步数取min即可。

总时间复杂度$\mathcal{O}\left(n + nlog^2n\right)$，足以通过此题。