Comet OJ 国庆欢乐赛（2019）题解

A. 轰炸平面镇魂曲

易知只有 5，6，8 三种情况，即长方形与xy轴均不相交，只与一个轴相交，与两个轴都相交。(注意长方形的边刚好在某一个轴上不算相交)。根据a * c与 0 的关系，b * d与 0 的关系，分类讨论即可。

时间复杂度O(1)

出题人代码

B. 卖萌鸡尾酒

先将五次搜索到的人数排序。设 A 为排序后的最大值，B 为剩下四个数字之和。因为只要有一个人在两个关键词中出现就被认为是“有缘人”。所以拿 A 与 B 每次拿出一个人两两匹配即可组成一个“有缘人”。如果 A 大于 B，则只能组成 B 个“有缘人”。

当 A≤B 时，可以组成(A + B)/2个“有缘人”。用数学归纳法证明：每次将最大的两个数字减一，组成一个“有缘人”，此时最大的数字仍然小于总和的一半，直至最后总人数小于二为止。

时间复杂度O(1)

出题人代码

C 两排房子

双指针显然是可以做的。直接派两根指针分别从两排房子的最左边出发，每次移动靠左边的指针，并判断是否有“对门”关系。

时间复杂度：$O(n+m)$

或者我们可以使用一个代码更加简洁的做法：

由于房子之间是不交的，则必有 $l_i < l_{i+1}$ 和 $r_i < r_{i+1}$。

使用 STL 中的 lower_bound 可以定位右端点$\geqslant l_i$ 和左端点 $\leqslant r_i$的位置，对于i来说，能够贡献答案的即为这两个位置之间的线段。

时间复杂度：$O(n\log_2 m)$

出题人代码

D1 入学考试(简单版)

简单版中的试卷数量 n 只有10^5，可以直接对做完试卷的数量直接进行枚举。剩下的时间贪心地去做耗时低的题目，由于可以不按顺序做题，所以需要先对题目按耗时排序，再用前缀和+二分法来算出剩下的时间能做多少题目。

时间复杂度：$O(m\log_2 m + n\log_2 m)$

出题人代码

D2 入学考试(困难版)

当困难版的试卷数量达到10^9时，显然枚举做完试卷的数量会超时。如何才能优化枚举呢？

假设做完试卷的个数为 Q，最终的得分为 ans。我们会发现随着 Q 的增加，ans 的变化可能并不满足三分的条件。如 ans—Q 的曲线是

一大段水平，中间只有一小段是极大值。那么三分很有可能搜不到极值。

我们重新对 D1 的枚举进行考虑。假设已做完的试卷为 Q，剩下的时间优先做耗时低的题目，并做到第 P 题时时间耗尽。假设 P 固定，我们会发现有很多的 Q 会满足“剩余时间做到第 P 题时时间耗尽”这个条件。并且这些 Q 是一个区间，可以设为$[Q_l, Q_r]$。我们发现在这段区间内，ans 的变化是一个一次函数。如果我们需要取 ans 的最大值，就会发现最大值一定在这个区间的两端点，即只需要去计算Q_l, Q_r对应的 ans 即可，而不需要去计算区间内部。

我们只需要枚举 P，求出它所对应的区间，再对这些区间的端点进行枚举即可算出答案。

时间复杂度 $O(m \log_2 m + m)$

出题人代码

E 不存在的粉丝群

由于是求最多的好友数，可以先把所有人都两两连边。然后再根据输入数据删边使得其满足题意即可。具体来说，当 k 为奇数或 k 大于 n 时，不存在答案。否则对 n 的奇偶进行分类讨论即可算出答案。

时间复杂度 $O(1)$

出题人代码

F 高速公路

不超速的情况：由于车速的变动幅度有限，因此并不一定每个点的速度都能达到限速那么大。所以我们可以先将每个点 i 真实可以到达的速度$Real_i$算出来。当某个点 i 满足$Real_i == limit_i$时（即第 i 点的限速），我们认为这个点为“关键点”。

对于非关键点来说，本身的速度就因为其他限制而无法到达限速。即使选择在这一点超速，也无法使得最后的“快速值”更大。所以如果选择超速，一定是在这些关键点超速。如果关键点超速，那么它周围的点也会随之变化。

求出每个点的实际速度$Real_i$的方法：可以考虑第 i 点会被三个 limit 所限制，一个是$limit_i$，一个是$Real_{i-1}+10$，一个是$Real_{i+1}+10$，这个点的实际速度是这三个 limit 的最小值。所以我们可以正反各扫描一遍求出$Real_{i-1}$和$Real_{i+1}$对此点的限制。

现在我们算出了第i个关键点的位置在pos_i。也就是说我们要枚举关键点，使得这一点的限速变为正无穷，重新计算“快速值”。但是如果不加限制地直接枚举，最坏复杂度也会是 n^2。但我们发现在枚举当前关键点时，上一关键点和下一关键点仍然满足$Real_i == limit_i$的限制。也就是说我们在枚举第i个关键点时，只需要对$[pos_{i-1}, pos_{i+1}]$重新判断车速即可。这样均摊时间复杂度是O(n)的。

时间复杂度O(n)

出题人代码

G 字符串

我们知道，一个子串是字符串的后缀的前缀。

这就很自然地让我们想到了后缀数组。

我们可以将原串 T 与询问的字符串$S_1, S_2 \ldots S_n$拼接构造出一个新字符串，

我们令新字符串为 $\mathrm{T}cS_1cS_2cS_3c \ldots cS_n\mathrm{c}$

其中 c 为大于字符串中所有字符的特殊字符

我们在这个串上建立后缀数组，可以得到文本串的每个后缀与匹配串之间的关系。

由于我们在每个询问串中加了 c，从这个询问串开始的位置的后缀的排名就是这个询问串的排名。

按照字典序由大到小枚举后缀，记 mini为目前在 T 中的后缀的最小位置。如果枚举到了某个 S 的开头，他的答案就是 mini。

要注意的是要保证这个串的终止位置也要在 T 中，否则答案就是 -1

时间复杂度：$O(siz_T + \sum siz_S)$.

出题人代码

H 二人桌游

由于我们是知道最终态的，即一个为 0，另一个为x(0 < x < n),我们可以先把这些点全部设为必胜态，那么假如有一个点只能到达这些必胜态的点，那么这些点就是必败点。根据题意知道状态(x,y)的转移关系是((x + y)%n, y)或者((x * y)%n, y)，我们就可以将所有这些关系

连边，用 BFS—拓扑排序的思想来解决这道题。

首先从这些已经确定状态的点开始 BFS 逆推，如果是后手必胜，那么推到的点直接进入广搜队列，状态是先手必胜。如果是先手必胜，那么先放着。如果两个点都是先手必胜推到一个点，那么就是后手必胜。否则一定是平局，因为所有环上的点都有环边，并且先手都不会主动走到必败点。

时间复杂度：$O(n^2)$

出题人代码