试题分析

Gromah

A. All with Pairs

首先，前缀的总数量和后缀的总数量都是 $O(\sum |s_i|)$ 的，所以可以先把每个后缀的 hash 值存起来。然后枚举每个串 s 的每个前缀 $s_{1..i}$，有多少个后缀和枚举的这个前缀相同，记为 Cnt[i]，则可以说明有 Cnt[i] 个串的后 i 位与 s 的前 i 位相同，但这个不能直接拿来算答案，因为一个串可能会给 Cnt[] 数组带来大于 1 的贡献。

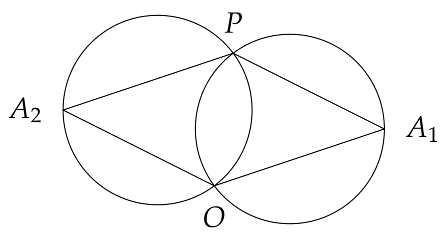
现在考虑去重，比如两个 “aba”，其中会有两个前缀 “a”, “aba” 被算到，但实际上只应该取 “aba” 而不是 “a”。所以我们考虑把每个串的 next 数组求出来，求完 next 数组之后再从前往后 Cnt[Next[i]] -= Cnt[i] 即完成去重，然后直接算答案即可。

时间复杂度：$O(\sum |s_i|)$

B. Boundary

大暴力是枚举两个点，再枚举其他点看在不在枚举的两个点和圆点所确定的圆上，这是 $O(n^3)$ 的。

但我们可以利用 “同弧所对的圆周角相等” 这一性质。具体来说就是先枚举一个点 P，然后枚举其他所有点 A，并计算 $\angle OAP$，那么我们找到这些角度的众数即可。不过这里有一个问题，就是角相等虽然可以说明弧相等，但这两段相等的弧可能不在同一个圆上，而在关于 OP 对称的两个圆上，如图：



而解决方法也很简单，只需规定 A 只能在 $\overrightarrow{OP}$ 的下方，即 $\overrightarrow{OP} \times \overrightarrow{OA} < 0$ 就可以了。

时间复杂度：$O(n^2 \log n)$

C. Cover the Tree

方法很多，这里说一下标程的做法。

首先设叶子节点个数为 s，那么答案下界就是 $\lceil \frac{s}{2} \rceil$。

如果 $n \leq 2$ 则答案显然。现在考虑 $n > 3$ 的情况：取任意一个非叶子节点作为根，然后求一个 dfs 序，并把所有叶子节点按 dfs 序排序，记为 $l_1, l_2, \cdots, l_s$。先假设 s 为偶数，那么所构造的 $\frac{s}{2}$ 条链为 $l_1 \to l_{\frac{s}{2}+1}, l_2 \to l_{\frac{s}{2}+2}, \cdots, l_{\frac{s}{2}} \to l_s$。

可以保证每条边都会被覆盖到，不妨设这条边的儿子节点所覆盖的叶子节点区间为 $[L, R]$：

- 如果 $R \leq \frac{s}{2}$，则这条边会被 $l_R \to l_{R+\frac{s}{2}}$ 覆盖
- 否则如果 $L > \frac{s}{2}$，则这条边会被 $l_{L-\frac{s}{2}} \to l_L$ 覆盖
- 否则因为根节点度数不为 1，所以 $L \neq 1, R \neq s$ 中必有一个是满足的，如果 $L \neq 1$ 则这条边会被 $l_1 \to l_{\frac{s}{2}+1}$ 覆盖，如果 $R \neq s$ 则这条边会被 $l_{\frac{s}{2}} \to l_s$ 覆盖

如果 s 是奇数，只需给根节点再接一个叶子节点，按照上述方法构造之后再把新加的这条边删掉即可。当然写代码可以不用这么写，这只是证明 s 是奇数的时候也是有下界解的。

时间复杂度：$O(n)$

D. Duration

把两个时刻在这一天的秒数算出来然后减一下即可。

时间复杂度：$O(1)$

E. Exclusive OR

首先考虑到当 $i > 19$ 之后有 $ans_i = ans_{i-2}$。简单证明一下：首先 $\forall i \in N^+, ans_{i+2} \geq ans_i$，只需要在 ans_i 的基础上随便挑一个数选两次就可以了，然后如果 $i > 19$ 之后 $ans_i > ans_{i-2}$，则说明至少需要 $i-1$ 个数才能达到满秩，而这里的 $i-1$ 至少为 19，但题目数据保证最大的秩为 18，这两者是矛盾的。

所以我们对于 $i \leq 19$ 每次用异或卷积算答案，$i > 19$ 之后令 $ans_i = ans_{i-2}$ 即可。有一组 hack 只求到 $i \leq 18$ 的数据：$1, 2, \cdots, 2^{17}, 0, 0, \cdots$，其中 $ans_{17} = 262142, ans_{19} = 262143$。

可能有个坑点，就是有某个 i 满足 $ans_i = ans_{i+1}$ 的时候，ans_{i+2} 及以后的值并不一定会等于 ans_{i+1}。一个反例：$5, 6, 24, 28, 30$，其中 $ans_1 = ans_2 = 30, ans_3 = 31$。

可能还有一个坑点，就是有某个 i 满足 $ans_i = ans_{i+2}$ 的时候，ans_{i+4} 及以后的值并不一定会等于 ans_{i+2}。一个反例：$4, 6, 12, 15, 1022$，其中 $ans_1 = ans_3 = 1022, ans_5 = 1023$。

时间复杂度：$O(n + w \log^2 w)$

F. Fake Maxpooling

暴力求 A 矩阵的复杂度是 $O(nm\log n)$ 的，大概是会 TLE 的，考虑如下 $O(nm)$ 的求法：

```
for (int i = 1; i <= n; i ++)
    for (int j = 1; j <= m; j ++)
        if (!Gcd[i][j])
            for (int k = 1; k * i <= n && k * j <= m; k ++)
                Gcd[k * i][k * j] = k, A[k * i][k * j] = i * j * k;
```

之后再单调队列就可以了。

时间复杂度：$O(nm)$

G. Greater and Greater

考虑 bitset，对每个 A_i 求一个长为 m 的 bitset S_i，其中 $S_i[j]=1$ 当且仅当 $A_i \geq B_j$。注意到本质只有 $O(m)$ 种不同的 bitset，具体就是把 m 个数排完序之后，第 i 个 bitset 就在第 $i-1$ 个 bitset 的基础上在第 i 大的数对应的位置多一个 1，所以预处理这些 bitset 复杂度是 $O(\frac{m^2}{W})$ 的。

再设 n 个长为 m 的 bitset，记为 cur_i，其中 $cur_i[j]=1$ 当且仅当 $\forall k \in [j,m], A_{i+k-j} \geq B_k$。可知有：

$$cur_i = ((cur_{i+1} >> 1) \mid I_m) \,\&\, S_i$$

其中 I_m 表示一个只有第 m 位为 1 的 bitset。

所以只要统计有多少个 cur_i 的 $cur_i[1]=1$ 即可，这就是最后的答案。写代码的时候可以不用把每个 cur_i 都存起来，只弄一个 cur 然后不断滚动就可以了。

时间复杂度：$O(\frac{nm}{W})$

H. Happy Triangle

首先，我们在回答询问的时候可以只考虑选大小相邻的两个数，因为对于一组 a,b,x，不妨设 $a \leq b$，如果能构成三角形的三边，那么我们完全可以把 a 调换成 b 的前驱并可以保证仍然是合法的。

所以我们只需要维护每两个相邻的数之间的差。然后对于一次询问 x，有两种情况：

- x 是最大边，那么找到 x 的两个前驱 p_1,p_2 并判断是否满足 $p_1+p_2>x$ 即可，这个用个 set 维护一下就行
- x 不是最大边，那么需要找到所有相邻的两个数，如果有一对相邻的数 $a,b\,(a \leq b)$ 满足 $b \geq x, a+x>b$ 则答案为 “Yes”，这个可能要写手写一个平衡树，每次去查一个后缀的最小值，其中每个点的权值就是其与前驱之差，这里可以加两个哨兵节点 -1 和 $+\infty$ 来避免边界讨论

时间复杂度：$O(q\log q)$

I. Interval

这个题本质上就是一个网格图网络流，把每个区间 $[l,r]$ 定到 (l,r) 的位置，其中源点即为 $(1,n)$，再设汇点为 $(n,1)$，并把所有 (i,i) 的点跟汇点连一条流量为正无穷的边即可，除此之外没有被限制的两点之间也要连流量正无穷的边。

这种问题一般是转对偶图然后变成求最短路，具体请参考题目“狼抓兔子”。

时间复杂度：$O(n^2 \log n)$

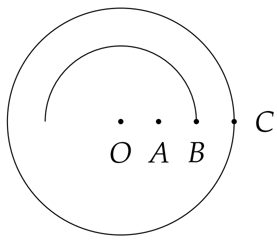
J. Just Shuffle

首先把 A 的所有环求出来，不妨设这些环的大小为 $r_1, r_2, \cdots, r_c$。因为 k 是大质数，所以可以对每个 i 求一个 $inv_i = k^{-1} \pmod{r_i}$，这样的话只需要把 A 中的第 i 个环转个 inv_i 次，则可以得到一个合法解。至于为什么要把 k 设成大质数，主要是要保证可以用上述方法做出来。事实上存在不能用上述方法做但确实有解的情况，比如 $n=4, k=2, A=\{3,4,1,2\}$，可知所有环的大小都是 2，有 $\{1,3\},\{2,4\}$ 两个环，而显然 $\gcd(2,2) \neq 1$，但这个数据是有解的，即 $\{4,1,2,3\}$，所以为了避免这种可能要拆环的情况本人就把数据范围设成那样了。

时间复杂度：$O(n)$

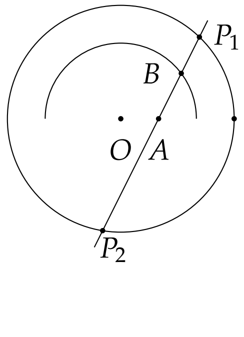
K. Keyboard Free

首先可以假设 A 是定点，B 只在某个半圆上运动，C 在整个圆上运动，如图：



然后关于 $S_{\triangle ABC}$ 的计算方法，一个比较直观的想法是借助点 O 利用叉积去算有向面积。然而这个做法的瓶颈在于不能简单用叉积算面积然后积分，因为叉积可能会算出负数，但本题是要求算正面积的，所以要处理符号，这是比较困难的，所以标程并没有算一个 $O(1)$ 的公式。

标程的做法是首先枚举 B 的位置，不妨设 B 为 $(r_2\cos\alpha, r_2\sin\alpha)$，取 l_{AB} 与 C 所在的圆的交点为 P_1, P_2，那么 C 到 l_{AB} 的期望距离是可以用积分算出来的，再乘以 $\frac{|AB|}{2}$ 则得到 B 确定的情况下 $S_{\triangle ABC}$ 的期望面积，这样只要对 B 这一维进行 $O(\frac{1}{eps})$ 的积分即可。如图：



时间复杂度：$O(T\frac{1}{eps})$