SOL

T1

设 V 为 $\max A_i$。

可以发现一个子区间的美丽度只有 $1,2,3$ 三种可能，因为 $\gcd$ 每次不变或至少除 2。

$\mathcal{O}(n^2)$ 暴力可以获得 12pts。

因为 $\gcd$ 每次至少除以 2 或不变，所以直接枚举左端点，找 $\gcd$ 相同的那些段，最多有 $\mathcal{O}(\log_2 V)$ 段，根据实现可以做到 $\mathcal{O}(n\log_2 V)\sim\mathcal{O}(n\log_2 V\log_2 n)$，可以获得 28pts，不过我想没有人写。

测试点 $8\sim 11$ 写一个值域 dp 即可，加上之前的可以获得 44pts。

特殊性质 A 是好做的，分类讨论即可。

特殊性质 B 可以写一坨道理，大概就是对暴力进行一些剪枝。

到这里为止就可以获得 68pts。

先考虑一下特殊性质 C，发现 2 不可能成为一个子区间的美丽度。

一个子区间的答案为 1 当且仅当这个子区间的 $\gcd\neq 1$，设 R_i 为满足左端点为 i， 右端点最远到 R_i 满足这个子区间的 $\gcd\neq 1$。

怎么快速求 R_i 呢，显然直接二分会 tle，考虑对每个数分解质因子，然后倒着扫，维护每种质因子当前能够延伸出去的最远位置，那么 R_i 就为 A_i 所包含的那些质因子 能够延伸出去的最远位置的 $\max$。

复杂度为 $\mathcal{O}(f(V)n)$，其中 $f(A)$ 表示 $1\sim A$ 中不同质因子种数的最大值，$f(5\times 10^6)=7$，可以接受。

到这里就有 80pts。

考虑正解，如何判断一个子区间的美丽度是否为 3?

设 R_{2_i} 为满足左端点为 i， 右端点最远到 R_{2_i} 满足这个子区间的 $\gcd\neq 1$（除去 2 这个质因子）。

设 b_i 为 A_i 所含 2 的质因子个数。

首先得出现 1 和 2，1 就是这个子区间的 $\gcd=1$（除去 2 这个质因子），难点在于判断出现 2，可以发现前缀 $\gcd$ 为 2 的右端点也是一段区间（如果存在），只需要找到这个区间的左端点。

一个符合条件的右端点条件有两个：

- 这个位置要在 R_{2_i} 之后。
- 这个子区间的 b_i 的最小值为 1。

开个指针倒着扫一遍即可。

时间复杂度 $\mathcal{O}(f(V)n)$，可以获得 100pts，不卡常数，时限为 std 的 2 倍。

T2

题目模型：给出N个点，由1至N编号。每个点有两个权值a和b。找出一个经过所有节点有且仅有一次的环，使得相邻节点的不和谐值的最大值最小。

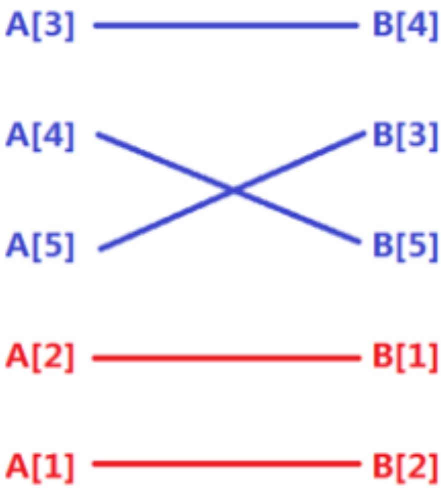
不和谐值定义如下：假如节点j是节点i在环中的直接后继，并且$a_i > b_j$，则不和谐值为$a_i^2 - b_j^2$，否则不和谐值为0。

模型简单清晰，有点像哈密尔顿回路的问题，但那是NPC问题，往那方面想只会徒劳。我们需要另辟蹊径，寻找突破口。

我们先观察不和谐值的定义：$a_i^2 - b_j^2$，我们先忽略最终要求形成一个环这一条件的话，我们要想不和谐值整体的最大值尽量小，最好的分配方案则是a, b数组分别从大到小排序，然后按最大A值与最大B值为相邻点，次大A值与次大B值为相邻点...最小A值与最小B值为相邻点，这样分配最终答案肯定最小值。这性质比较明显，在此就不多作证明。

但是，假如我们按照这样的顺序分配每个点的后继，那么，就会出现形成若干个环的情况。现在我们要寻找一种方法，使得这若干个环连在一起，并且能够使得最大值尽可能地小。

我们把在同一个环上的点染成相同颜色，不同环上的点染成不同颜色。对样例进行这种操作，如左图。



假如，我们现在想把蓝色环和黑色环合并的话，最优的情况是将5变成4的后继，3变成5的后继。如右图。

由此可以发现，按照从大到小排序后，假如第i大的A与第$i+1$大的B不是同一颜色的，我们可以将第$i+1$大的B作为第i大A的后继，而且，第i大的A与第$i+1$大的B所产生的不和谐值是修改过程中不和谐值最大的。（证明略）

至此，我们可以得出一种较好的合并两个不同环的方法，而这种方法只需要关注第i大的A值与第$i+1$大的B值。再仔细分析，这种模型貌似似曾相识。其实，这里我们很容易联想到最小生成树。把一个环看做一个点，把这种合并两个环的方法看做一条连接两个点的边，现在问题就变成，找一棵生成树，使得边的最大值最小。

值得注意的是，最终答案未必是生成树上的最大边，或许是排序后分配的后继，因此，最终答案还需要考虑排序后分配后继时所产生的不和谐值的最大值。

总结一下，其实这算法实现起来很简单，先把A,B数组排序，然后初始成若干个环，然后在转换成经典的生成树问题。时间复杂度$O(n \log n)$。

T3

n 个点，点权是 0 到 $2^m - 1$ 的随机值，边权是点权的 XOR，询问最小生成树的期望大小。

30% 算法：暴力一波。

50% 算法：如果有两个点权值相同，显然可以合并，那么我们只需要对于 2^{2^m} 种点权出现状态考虑就好了。

100% 算法：

令 $f(n, m)$ 表示 n 个点，点权是0到 2^m—1 的随机值时所有情况的最小生成树的和，只要除以 2^{rm} 就是答案了。

专虑如何求最小生成树，根据第 m 位是 1 还是 0 将点分成 S 和 T 两个集合，那么 S 和 T 内部一定形成生成树，然后在 S 到 T 找到一条最小的边就可以了。

用 $g(S,T,m)$ 表示一边有 S 个点，一边有 T 个点，点权是 0到 2^m—1 的随机值时所有情况的最小边的和，那么通过枚举有多少个 1 可以发现

$f(n,m)=\sum_{i=0}^{n}\binom{n}{i}(2^{(n-i)(m-1)}f(i,m-1)+2^{i(m-1)}f(n-i,m-1)+g(i,n-i,m-1)+2^{(n+1)(m-1)})$

那我们的问题就是如何求 $g(S,T,m)$。

直接求 $g(S,T,m$ 可能不大好求，可以令 $p(S,T,m,k)$ 表示一边有S个点， 一边T个点，点权是0到2^m—1的随机值时最小边大于等于k的情况数，那么

$g(S,T,m)=\sum_{k=1}^{2^m-1}p(S,T,m,k)$

考虑如何求最小边，还是根据第 m 位是 1 还是 0 将点分成 S_0,S_1,T_0,T_1 集合， 肯定是 S_0 和 T_0 或者 S_1 和 T_1 里的边，于是可以递归下去。那么求p的方法也是类似求f，枚举 S 里多少个 1 和 T 里多少个1。

时间复杂度：$O(n^4 2^m)$，复杂度较大但是常数很小。

T4

分析一下问题,可以发现瓶颈在于限制牛仔序列

前$20pts$似乎直接状压+子序列DP即可

$sub3$是个很好的启发,由于A就是牛仔序列,我们只需要统计每个A的贡献

直接枚举A的位置算方案数,答案是$(n-m+1)k^{n-m}$

$sub4$我们先剔除$m=n$的情况

剩下的,考虑容斥,先计算所有A可能的贡献$(n-m+1)k^{n-m}$,最后减去没有牛仔序列的序列中A的贡献

注意到没有牛仔序列等价于不存在连续的长度为K元素不同的段

同样考虑子序列dp

设$dp_{i,j}$表示填前i个数后末尾有长度为j连续的元素不同的段的方案数

转移的话考虑

$j\to j+1$,即填上前面j个没出现的数,转移系数为$k-j$

$j\to p,(p\le j)$,即填上前面出现的数,根据与末尾的位置决定p

这里直接后缀和优化即可,这里我们只需限制$j<k$即可保证无牛仔序列

注意这里计算的是牛度，我们可以发现每个元素均是等价的，所以我们可以计算所有长度为m元素不同的段的贡献最后除$\frac{fac_k}{fac_{k-m}}$,对于这个的计算,可以记录一个辅助数组表示牛度,转移是一样的,就是最后$j\ge m$时要算上自己的贡献

其实这里离正解就不远了,可以发现就只剩一种情况:A中有相同元素

如果A有相同元素,一个合法的牛仔序列一定不会横跨A,因此我们可以考虑直接枚举A的匹配位置算方案数

可以发现这个可以借助上面的dp的定义和转移,唯一的区别在于初值不同

时间复杂度$O(nk)$