T1 Static Query on Tree

解法1：

对于每次询问，将A和B中所有点到根路径打上标记A/标记B，C中所有点将子树打上标记C，统计有多少个点同时打上了三种标记

具体而言可以这样做：

树链剖分+线段树

打A标记时采用区间覆盖和区间求和，维护区间内有多少个点打上了A标记，记作sumA

打B标记时，当整个区间都要打上B标记时，可以通过sumA算出有多少个点同时被A和B标记，这样就能维护区间内有多少个点同时被A和B标记

打C标记的时候类似

时间复杂度$O(nlog^2n)$

解法2：

考虑最简单版本的问题1，有一个集合A，选A中的点走到根，数有多少个点会被走到。

可以将A中点按照dfs序排序，将A中点的深度加起来，然后减去相邻两点lca的深度和。

时间复杂度$O(|A|logn)$

再考虑一个稍复杂的问题2，现在有两个集合A和B，对A和B都求一个问题1，然后求交集，问交集大小。

直接算交集比较复杂，可以采用容斥原理，将A，B，A∪B的答案都算出来，然后容斥出A∩B的答案。

时间复杂度$O((|A|+|B|)logn)$

这样原问题其实就是将问题2在若干个子树上面做，但是如何保证复杂度呢

对于集合C，可以发现，如果有一个点是另一个点的后代，那么将后代从集合C里去掉并不会影响答案。

去点的过程具体如下：

将C中所有点按照dfs序排序，每次加入一个点时看是否在之前点的子树中。

这样最后保留下来的子树两两不交，这意味着每次只需要取A中的一段，且段之间两两不交。回答一次询问的复杂度为$O((|A|+|B|+|C|)logn)$

时间复杂度$O(nlogn)$

T2 ShuanQ

$P\times Q-1=k\times M$，其中k为一个正整数，将右边进行质因数分解，最多只有一个质因数满足大于P和Q

可以考虑反证，假如M_1和M_2都大于P和Q，那么$M_1 \times M_2 > k \times M$，矛盾

如果大于P和Q的质因数存在，那么它就是M

T3 Alice and Bob

将每个值为i的数字视作2^{n-i}，Alice获胜条件就是最终局面中能出现2^n

Alice将数字分成两个集合，Bob将其中一个集合减1，就相当于将这个集合中的数全部乘2，然后将另一个集合删去。

如果黑板上的数字总和大于等于2^n，由于所有数字都是小于2^n的2的幂次，那么Alice的每次分割总能使得两个集合的值均不小于2^{n-1}，这样一来黑板上数字总和总是大于等于2^n的，且数字总数在减小，Alice必胜

如果黑板上的数字总和小于2^n，Bob只要每次都删除总和较大的那个集合，黑板上数字总和会一直减小，不可能达到Alice胜的局面，Bob必胜

因此，将每个值为i的数字视作2^{n-i}后，判断所有数的总和即可

时间复杂度$O(n)$

T4 Dusk Moon

对于一个点集，它的凸包覆盖住了所有点，且最小覆盖圆覆盖住了凸包，因此仅保留凸包的顶点不会影响答案。 由于点的坐标在给定的正方形范围内随机，因此一个点集的凸包的期望顶点数为$O(logn)$，使用线段树直接记录区间凸包点集，然后对于$O(logn)$个点运行最小圆覆盖算法即可。时间复杂度$O(qlog^2n)$。

T5 Bowcraft

令 $dp[i]$ 表示从 0 级升级到 i 级期望的数量。

假设现在等级为 i，买了一本书 (a,b)，记 $\alpha = \frac{a}{A}, \beta = \frac{b}{B}$ 。

若使用这本书，升到 $i+1$ 级的期望是 $dp[i] + 1 + (1-\alpha)(1-\beta) \cdot (dp[i+1] - dp[i]) + (1-\alpha)\beta \cdot dp[i+1]$

若不使用这本书，升到 $i+1$ 级的期望是 $dp[i+1] + 1$

得到 dp 方程：

$$dp[i+1] = \frac{1}{AB}\sum_{a,b}\min\{dp[i+1]+1, dp[i]+1+(1-\alpha)(1-\beta)\cdot(dp[i+1]-dp[i])+(1-\alpha)\beta\cdot dp[i+1]\}$$

对于当前的等级 i 和一本书 (a,b)，如果要使用这本书，那么升到 $i+1$ 级，不使用的期望 $\geq$ 使用的期望。

即 $dp[i+1]+1 \geq dp[i]+1+(1-\alpha)(1-\beta)\cdot(dp[i+1]-dp[i])+(1-\alpha)\beta\cdot dp[i+1]$

化简，得 $dp[i+1] \geq dp[i]\cdot\frac{\alpha+\beta-\alpha\beta}{\alpha}$

从这个式子可以看出，一本书的 $\frac{\alpha+\beta-\alpha\beta}{\alpha}$ 越小越容易被使用。即 $\frac{\beta(1-\alpha)}{\alpha}$ 越小越好。

将所有书按照 $\frac{\beta(1-\alpha)}{\alpha}$ 从小到大排序，枚举所有的 t，只使用 $\frac{\beta(1-\alpha)}{\alpha}$ 值前 t 小的书，其他书不使用。则

$$AB\cdot dp[i+1] = (AB-t)\cdot(dp[i+1]+1) + \sum_{a,b(\text{前}t\text{小})} dp[i]+1+(1-\alpha)(1-\beta)\cdot(dp[i+1]-dp[i])+(1-\alpha)\beta\cdot dp[i+1]$$

化简，得 $dp[i+1] = \dfrac{AB+dp[i]\cdot\sum\limits_{a,b(\text{前}t\text{小})}\alpha+\beta-\alpha\beta}{t-\sum\limits_{a,b(\text{前}t\text{小})}1-a}$

枚举所有的 t，得到 $dp[i+1]$ 的最小值，递推可得 $dp[K]$。复杂度 $O(KAB)$

T6 Two Permutations

先特判S中所有数字出现的次数是否不都为2，此时答案为0。

设$f[i][j]$表示P的前i项匹配上了S，且P_i匹配S中对应数字第j次出现的位置时，方案数为多少。因为数字出现次数都为2，因此状态数为$O(n)$。转移时枚举P匹配的位置中间的那P_{i+1}段连续子串需要完全匹配Q中对应的子串，使用哈希$O(1)$判断即可。

总时间复杂度$O(n)$

T7 Link is as bear

思维+结论，可以证明，从这n个数里任取一些数异或起来的方案，都是可以构造出对应的操作来做到的。

所以，问题完全等价于给n个数，从中选一些数，使得这些数的异或和最大，这是线性基的经典应用。

证明如下:

- 1、如果序列里有连续的两个0，那么一定都可以构造出操作方案。例如$0,0,a_1,a_2$中，比如我们要保留a_1不保留a_2，就可以先两次操作变成$a_1,0,0,a_2$，再两次操作变成$a_1,0,0,0$，因为上述操作总可以保证操作完后还有两个0，所以可以一直往下处理。如果两个0两边都有数字也没问题，先处理完一边的再回过头来处理另一边。
- 2、如果用1和0表示最终想保留/删除一个数字，则出现110或011的形状时，可以通过操作删掉想删除的那个数字，并构造出两个连续的0。以110为例:

  $a_1,a_2,a_3 \to a_1\oplus a_2, a_1\oplus a_2, a_3 \to a_1\oplus a_2, 0, 0$
- 3、由1和2可以看出，只要出现了110或者011就可以任意保留数字，而有连续两个0也可以任意保留数字。因此，只要有连续两个0或者连续两个1的序列，都可以构造出方案。

- 4、现在只剩下01交替出现的情况不能构造方案了，但是条件中说会有一个数字出现在两个位置，不妨设为x。假如最终的答案包括x，那么可以将一个x视作0，另一个视作1；假如最终的答案不包括x，那么两个x同时视作0或者同时视作1，这样序列就会出现连续的0或者连续的1了。

T8 Link with Running

图论。整体思路：在最短路图上跑最短路。

- 先用dijkstra在第一个权值上面跑出最短路图（最短路图的边权为第二个权值），求出第一个答案。
- 注意到因为第一个权值的范围内包含0，因此最短路图不一定是个DAG，需要用tarjan将强连通分量缩起来，在缩点时，内部的边的第一个权值一定都是0，因为保证了答案一定存在，因此第二个权值一定也都是0，不然就出现正权环了
- 在DAG上求最长路即可