A: 水题

水题，直接输出即可

B: 动态规划

将买卖零次和一次全部转化为买卖两次，相当于 $a \leqslant b \leqslant c \leqslant d$。题目等价于枚举 $a \leqslant b \leqslant c \leqslant d$，使得(s[b]/s[a])*(s[d]/s[c])的值最大。

定义 dp1[i]表示 i 及其左边的最小数，dp2[i]表示 i 及其右边的最大数。那么，s[i]/dp1[i]代表以 b=i 时，最大的(s[b]/s[a])，dp2[i]/s[i]代表以 c=i 时，最大的(s[d]/s[c])。在此基础上，定义 dp3[i]表示 i 及其左边的最大(s[b]/s[a])，dp4[i]表示 i 及其右边的最大(s[d]/s[c])。枚举 i，求 dp3[i]*dp[4]的最大值。时间复杂度 O(n)。

C: 图论、有向图判环

题目中说明，当且仅当等待出现闭环时会出现死锁，我们可以将每一项进程看作图的一个点，一个等待（T1 W T2）看作一条 T1 到 T2 的有向边，则此题就是判断有向图是否有环。

对有向图判环有两种方法，一是拓扑排序，若能通过拓扑排序找到一个序列，则该图无环。还有一种是深搜，当搜索一条路经遇到一个在栈中的节点则有环。

D: 数论、互质数

定义 d[i]为 b[i]-a[i]+1，如果 d[1]~d[n]互质，则一定 AC，否则存在无法 AC 的可能（证明不作详述）。判断互质的方法有两种，一是对任意两个数判断 gcd 是否为 1，时间复杂度 $O(n^2 \log n)$，二是对每个数进行素因数分解，判断是否出现相同的因数，时间复杂度 $O(n\sqrt{d})$，其中 d 为 d[i]的数值范围（约为10^9数量级）。本题还有 O(nlogd)的做法，但需要使用 Java 中的 BigInteger 类。

定义 s=d[1]*d[2]*…*d[n]，则期望为 $20\times\frac{0+1+2+\cdots+(s-1)}{s} = 10s - 10$

E: 数据结构 / 思维

连续 xor 值为 1 的子区间，即存在奇数个 1 的子区间。利用线段树维护区间右端点向左延申的包含奇数个 1 的子区间数目。询问时，利用线段树将区间拆分，考虑每个小区间对答案的贡献，把加起来即可。

本题也有 O(n)的做法，统计每个 1 左边出现的连续的 0 的数量，记为这个 1 的权值，对于所有第奇数次出现的 1，建立一个权值前缀和数组，对于所有第偶数次出现的 1，建立另一个权值前缀和数组。对于每个询问的 R，如果该位置是 1，判断是第奇数次还是偶数次出现，并在对应的权值前缀和数组中求 L 到 R 的区间和。如果该位置是 0 则找到它前面的 1，类比上述操作。该方法需要考虑较多的边界条件。

F: 思维

两种做法，第一，对于每个位置，统计每个字符出现的次数，最后将每个位置出现奇数次的字符拼接起来就可以。第二，将所有字符串进行逐位异或，即可得到答案。

G: 数学、公式推导

$$\frac{1}{2^n-1}\times 1+\frac{2}{2^n-1}\times 2+\frac{4}{2^n-1}\times 3+\cdots+\frac{2^{n-1}}{2^n-1}\times n=n-1+\frac{n}{2^n-1}$$

H: 数论、莫比乌斯反演

当点不在边缘上时，过该点作三条边的平行线，将三角阵切分为三个平行四边形和三个正三角形。对于 n*m 的平行四边形，在顶点能看到的点数等于[1,n]和[1,m]内互质数的对数+2，对于 n 阶正三角形，可以通过 n*n 的平行四边形推导出对应答案，将这 6 块的答案相加，再减去重复计算的 6 个点。当点在边或顶点上时，类比上述方法。计算互质数对数可用莫比乌斯反演实现。

I: 字符串匹配、后缀数组

后缀数组，将所有每一行输入的数视为字符串，将所有输入的字符串拼接，并且记录下每一个原字符串的首字符在新字符串中的位置，这样原问题就转化为求两个字符串 a,b 的最长公共前缀是否为第二个字符串 b 的长度。

J: 几何

如果该点不在三角形的边上，则该点和三角形任意两个顶点一共可以组成三个三角形，如果这三个三角形的面积和等于原三角形面积，则点在三角形内。