构造

321C

- 点分，层数越小字母越大，经过点分中心的值都会比它小。
- 而点分的层数不超过 O(log n)，所以可以。

1450C1

- 考虑关键点是所有点的做法。
- 如果所有点都是关键点，显然有 3 种简单的染色且两两染色间没有重复点，即染(i+j)%3=0/1/2，那么关键点的方案相当于和它们求交，必定有一个不超过 k/3。

570D

- 在线：相当于在深度为 b 的点中，求 dfs 序在某个区间内的异或和；
- 离线：记录进入前当前节点询问的深度的异或值，做完子树后再异或。
- 每个点只有一种深度，所以复杂度正确。

1217D

- **只有编号小的连向编号大（或反过来）的边的图一定是 DAG。**
- 没有环可以构造答案为 1，有环显然答案就至少是 2，将从小的连向大的涂一种颜色，大的连向小的涂一种颜色。

1485D

- **lcm 不超过值域限制。**如果 k 可以为 0，那么所有数都填 lcm 即可。
- 黑白染色，所有限制都为黑-白，给黑点加上 v[i][j]^4

1365F

- **寻找不变量，和为 n+1 的位置构成的无序对不会改变。**先盲猜满足此条件即可。
- 从中间到两边确定，如果和当前 b_x,b_y 相同的是 a_i,a_j，即…a_i…a_j…,可以变成 a_j……a_i，（a_i…...a_j）再变成最后答案，故猜测成立。

1360H

- x=(2^m-n)/2，从小到大第 x 个数即为中位数，从小到大枚举 n 个数并更新中位数的值即可。

796D

- 最多删去 k-1 条边，说明分成 k 个连通块。
- 每个连通块恰好一个警察局，任意构造即可

1333D

- 求最少和最多的次数，中间的都可以求出。
- 最多的步数是每一次删除一个逆序对，最小的步数可以贪心。
- 再根据最小步数的方案往大调整。

1438D

- 操作后**异或和不变**。
- 若 n 是奇数, 设异或和为 x, 只能把所有数都变成 x。操作(1,2,3),(1,4,5),(1,6,7)…, 可以位置 1 变成 x, 且 2=3,4=5…, 接下来再做一次所有操作即可。
- 若 n 是偶数，最终异或和为 0，先判断无解，否则删掉最后一个做 n 为奇数即可。

543B

- 相当于你要钦定两条路，一条是 s_1 到 t_1，一条 s_2 到 t_2，最后没被删除，其它的都可以被删除。
- 但是这显然不可能，你不能枚举每条路径。不过你发现，你删除的是不在两条路径上的边，你只要枚举两条路径的交（一定是一条路径），剩下相当于两条路径独立了。
- 不过关键在于边权都是 1，那么你一定都选的最短路。你枚举两条路径交的端点 u,v(u<v)，考虑 s_1 到 t_1 和 s_2 到 t_2 是顺序经过 u,v 还是顺序经过 v,u，算一下最短路，就可以知道剩下的最多公路数量了。(其它部分有重复的话一定是更劣)

1366E

- 考虑如果知道了 l_1,l_2,…,l_k（即每个区间的左端点），那么需要满足 min(l_i,l_i+1,…l_{i+1}-1)=b_i。
- 而因为 b_i 单调递增，故可以改为 min(l_i,l_i+1,…,n)=b_i。
- 对于每个 i，对应的区间一定不相交，乘法原理乘起来即可。

1364D

- 题面**说一定有解**，那么可以选出一个 k 个点的导出子图。
- 看有没有环，如果有就输出，没有的话就是一个森林，黑白染色后在较大的一部分输出 k/2 个点即可。

1110E

- **差分，变成交换相邻两个 d**，排序后判断两个是否相等即可。

675C

- 相当于分成尽量多的段，每段和为 0，那么答案是 n-段数。
- 即找到一个 v，使得前缀和为 v 的数量最多。map 存一下即可。

1479B2

- 枚举 i，维护另一个序列的右端点 j。考虑新增的位置的数和 i,j 是否相同。
- 相同：那么答案不变是最优的。不同：考虑 i 和 j 下一次出现位置，**较大的替换掉**（保留较大的不如之后再换）。

1560F2

- 每次找到满足条件的最长前缀，在下一个位置+1，之后的位置都设成 0。

1332E

- 由于你可以给每个格子不停的+2，所以可以将初始的 a 都 mod 2，之后都操作都**在 mod 2 意义下**。
- 可以把操作从将相邻两个数同时+1 改成任意两个数同时+1（找一条这两个数间的路径，相邻都做一次操作）。
- 若 n*m 为奇数，那么 0/1 一定有一种个数是偶数，操作即可。(tot)
- 若 n*m 为偶数，如果是两个偶数则也可操作，两个奇数则无法操作。(tot/2)

1322B

- 考虑依次确定每一位的值，两个数加起来第w位是 1 当且仅当两个数的后k位之和满足 $2^k \le a_i + a_j < 2^{k+1}$

  或$2^{k+1} + 2^k \le a_i + a_j$。双指针算一下看奇偶性即可。到下一位的时候**基数排序更新顺序**。

1406D

- 不妨把 c **全部取个负**，变成 b,c 都不下降序列，a_i=b_i-c_i。现在答案为 max(b_n,-c_1)。
- 两个序列都要不下降，不妨 a,b,c **差分**一下。现在要求 a_1=b_1-c_1，A_i=B_i-C_i(i<=n-1)，B,C 非负。
- 答案是 max(b_1+B_1+B_2+…+B_{n-1},-c_1)。可以发现 B 之和最小为非负的 A 之和，设其为 k。
- 现在变为 max(b_1+k,-c_1)，分析一下就是 max((a_1+k+1)/2,k)。操作只会修改两个位置，更新一下 k 即可。

1392E

- 若路径的和为 x，把 x 二进制下的 0/1 分成若干个连续段，如果段切换则走到下一行，如果不是则往右走。
- 这个显然可以构造出来：若是奇数行则全为 0，若是偶数行则是 2^(i+j-2)。
- 显然对于同个 x 不存在多种方案，求方案就模拟第一行的过程。

1450C2

- 如果所有点都是关键点，显然有 3 种简单的染色。把**点分成(i+j)%3=0/1/2 三组**。
- 可行的染色方案：一组为 O，一组为另一种 X，剩下的一组不用改变。
- 求关键点的方案可以是它们求交，三种总操作次数为 k。根据鸽巢原理，必定有一个不超过 k/3。

1419E

- 观察一下样例，n=pq（p,q 是两个不同质数）需要加数，其它情况好像都不要加。
- n=pq 比较简单，其它情况大概是 p_1p_m,p_1 倍数,p_2p_1,p_2 倍数,…,p_mp_{m-1},p_m 倍数。

1437E

- 先使用经典套路，**第 i 个位置-i 变成单调不下降序列**。
- 考虑能同时保留的位置，那么一定是单调不下降的位置，每段里求一下 LIS（不下降）即可。

1470D

- 根据经典套路**先取一棵生成树**，如果取不出则原图**不连通显然不合法**。
- 否则按照 dfs 序一个一个加点。
- 如果之前加入过的相连点有染色，则它不能染色，且已经连通。
- 如果之前加入过的相连点都没有染色，则它必须染色，且依然会连通。

1553E

- 有 m<=n/3 的条件，交换一次最多使得 2 个数归位，那么方案若可行**至少有 n/3 个数在原位**。
- 对于每个位置，只有一个 k 使其在原位。那么只有最多 3 个 k 是可能可以的，对于这些 k 暴力判断即可

1338C

- 打表。可以发现 a 在[2^{2i},2^{2i+1})中依次增加，而对应的 b,c 在[2^{2i+1},2^{2i+2})中。
- 进一步发现相邻四个 b 满足 0231 的关系，c 满足 0312 的关系。
- 再进一步发现对于 a 在一个块内的，b,c 先分成四小块，大小顺序满足 0231,0312 再递归下去分。
- 确定了 a 在哪一个块之后就可以暴力了。

1305E

- 考虑**先构造最大的情况**，固定了 k 的话，那么由于前面只有 k-1 个数，最多配成(k-1)/2 对。
- 达到这个上界可以构造 a_i=i，找到最前面的位置 x，使得按照 a_i=i 构造，到当前位置的值就>=m 了。
- 现在要增加 x 位置的值使得 m 减小，可以直接暴力。
- 至于后面的位置，要让它们不影响答案。设当前填过的最大数为 v，那么从后往前，每次填的数-(v+1)即可。

1373E

- 考虑 x+k 进没进位。
- 如果没进位，枚举最后一位，之前的位贪心（尽量填 9）。
- 如果进位了，枚举最后一位和之前连续的 9 的长度，计算一下，再贪心（先填一个 8，再尽量填 9）。

1530E

- 如果只有一种字符，那只有一种方法，直接结束。
- f 为 0：如果有某种字符只出现一次，那么拿最小的只出现一次的字符当开头，其它的从小到大往后排（不可能拿出现果不止一次的字符，那会使得 f 至少为 1；也不可能取更大的字符，那字典序更大）。
- f 为 1：现在所有字符都出现至少两次（这种情况答案一定不为 0，不妨假设 a 出现了），想让字典序尽量小，先考虑有没有可能是以 aa 开头，那么之后就不能有两个相邻的 a（否则 f 至少为 2），只能形如 aababa…ca…。可以发现 a 的个数最多为其它数字个数+2，即 cnt>=n-cnt+2，cnt>=n/2+1。
  如果不满足条件（不妨假设 b 出现了），考虑开头为 ab，那么之后就不能再出现 ab，如果只有两种字符，那么 b 一定要出现在 a 前面，即 abbb…baa…a；如果有更多字符（不妨假设 c 出现了），那么可以为 abaaa…acbbb...bccc…c…，即 ab 之后填所有的 a，再填 c，剩下的从小到大填。

1254B2

- 只需要考虑 k 是质数的情况。
- 其实 **a_i 和 a_i mod k 是<u>完全</u>等价的**（不会出现负数啥的情况），那么**前面往后移和后面往前移也是等价的**，计算一下答案即可。

763B

- 根据四色定理显然有解。
- 所有矩形的边长均为奇数，可以根据**矩形左下角纵横坐标的奇偶性来把矩形分为四类**，分别染不同颜色。

1520F2

- **每 w 分一块**，先询问出前 i 个块中 1 的个数。
- 每次询问先在大块上二分，不消耗次数，再在小块上二分，需要 log w 次，用树状数组更新大块的前缀个数。
- 交互次数是 n/w+t*logw，w=8 就可以过。

1166E

- 看上去很难下手，先找点一定不合法的情况。
- 可以发现**两个限制没有交的话显然不合法**。
- 如果都有交，我给每个区间乘上一个不同的 p_i，那么区间的 lcm 是所有 p 之积，而补集一定没有 p_i，更小。

1451E2

- 先假设序列为 0,a[1]^a[2],a[1]^a[3],...，实际的序列为每个位置异或上一个数 v。
- 只需要确定任何一个位置的值就可以求出原序列。

- 如果现在的序列有两个值相同，则问一下它们的 and 就可以知道它们真实的值。
- 如果没有相同，说明 0,1,…,n-1 都出现了一次，找到现在序列的 0 和 n-1 的位置，它们对应实际的值是 v,(n-1)^v，再找另一个位置 x，询问一下这两个位置和 x 的 and，or 起来就是 x 位置真实的值。

576C

- 按照**莫队的方法**奇偶排序即可。

1486E

- w 很小，直接**拆点**。
- 对于第一步连出去的花费是 0，u_0 连到 v_w。
- 对于第二步连出去的花费是(i+w)^2，u_i 连到 v_0。

1296F

- 显然要满足 u,v 路径上的边权都>=w。
- 路径取 max 后判断是否合法即可。

1384B2

- 称无论什么时候在当前位置都是安全的点为安全点，否则为不安全点。显然不安全点能走的时间是一些区间。
- 若当前点为安全点且下一个点也是安全点，直接走过去即可；
- 若当前点为安全点且下一个点不是安全点，则在最优的时间走过去（递减且正好能走过去）。
- 若当前点不为安全点且下一个点是安全点，直接走过去即可；
- 若当前点不为安全点且下一个点不是安全点，则只要能走过去且不死就走过去，一直不走必定会死，**能跑路就赶紧跑路**。

549G

- 继续寻找不变量，**a[i]+i 总是不变**的。
- 初始的 a[i]+=i，条件变为：a_{i-1}>=a_i 的话交换 a_{i-1},a_i
- 如果有相等的显然最后会一直交换，否则最后会排序，再-i 输出即可。

766E

- 常规套路**枚举每个二进制位**。
- 记 dp[u][0/1]表示节点 u，路径异或值为 0/1 的路径数量（一段在 u 一段在子树内）。

715B

- 先把 0 边都设成 1，dijkstra 求出最短路，如果**>L 则显然无解**。

- 否则，你可以使得 0 边造成的最短路尽量变为原来+L，即如果 u 现在的最短路+边权<v 原来最短路+L，则扩大边权使得变为原来最短路+L。
- 最后看一下是否合法，如果不合法就无解。

1494D

- LCA 的权值为最大值的，说明它们 LCA 为根。
- LCA 的权值不为最大值的，说明它们在根节点同一子树。
- 递归下去即可。

1495C

- 切比雪夫距离>1 说明两个点**不是八连通**的。
- 把某一列都填 x，那么左边一列/右边一列的 x 都会和它连通。
- 但你显然不能每隔一列就都填 x，这样可能会这全是 x 的两列有多种连通方案。
- 考虑每隔两列都填 x，那么你要连接这全是 x 的两列，可以发现把第一行或第二行连起来即可。
- 如果 m mod 3 不等于 1，那么填 2,5,8,...列，否则填 1,4,7,...列。

780E

- 经典套路**先求个生成树**。
- 要求**每个点经过至少一次**，且总顶点数在 2n 级别。
- 考虑**欧拉序**，长度为 2n-1。直接将欧拉序分成均匀的 k 个部分即可。

1404C

- 如果询问的是整个序列，那么每次删除最后一个满足 a_i=i 的位置即可，把删除顺序记作 b_i。
- 询问的不是整个序列，找第一个 b_i<x，在之前的 b_i<=y 的都对答案产生了 1 的贡献，计算即可。

128B

- 建出后缀树，根据子树串个数之和判断往哪个子树走。

1415E

- 相当于**分成最多 k+1 个序列**，每个序列再按照顺序计算。
- 可以贪心考虑，显然分出的每个序列都是从大到小排序最好。
- 用堆维护每个序列当前的和，找到最大的序列加上当前的数。

788B

- 它要求不同的路径是仅走一遍的两条边不同。

- 把原图每条边看成两条，转化为删去两条不同的边，使原图中**存在欧拉（回）路**。
- 如果不连通答案为 0，否则条件是所有的度数都为偶数或恰好有两个点度数为奇数。
- 有三种情况：删两个自环；删一个自环和任意一条边；删两条有公共点的边。

888G

- 考虑 kruskal，对所有点的权值建 trie。
- 找到最高的 0 和 1 都出现的位，显然跨越两子树的边只有 1 条。
- 枚举点数较少的一侧的点，在字典树上查询异或最小值，然后递归处理两子树。

1526D

- 根据经典套路，**存在一种最小代价最大的方案，相同的数字都排在一起**。
- 4! 枚举四种字母的顺序，树状数组求值即可。

1003E

- 先搞一条长度为 d 的链。
- 现在要在链上挂点，挂的点的深度最多是 d-当前点到直径两端点路径长度的最大值。

1408F

- n=2^k 容易通过 k*2^(k-1)次来使得所有数相同。
- 到最小的 k 使得 2^(k+1)>=n，先将[1,2^k]变相同，再将[n-2^k+1,n]变相同即可。

985D

- 两张可能。
- 一种是 x,x-1,...,1 和一个单独的数 t(1<=x<=h,1<=t<=x，也可能没有)。
- 一种是 h,h+1,...,x-1,x,x-1,...,1 和两个单独的数 p,q(x<=h,1<=p,q<=x，也可能只有一个或没有)。
- 二分或解方程即可。

487C

- 设 b 为 a 的前缀积，若 a_i=n 则 b_i,...,b_n=0，所以 a_n=n；若 a_i=1 则 b_i=b_{i-1}，所以 a_1=1。
- 再分析一下发现对于 **n>4 且 n 不是质数的时候(n-1)!=0 mod n**，说明无解。
- 质数可以构造前缀积为 1,2,...,n，求逆元即可。

840B

- 按照经典套路**先取一棵生成树**。
- 有一种想法是从下到上，如果当前的度数合法则不取和父亲的边，否则取。

- 这样的话，除了根节点的度数奇偶性一定是对的，而度数总和一定是偶数，那么根节点的也确定了。
- 如果所有点的度数都确定且和是奇数显然无解。否则如果有-1 将它放到根节点即可。

1537F

- 按照经典套路，每次加或减的都是一个偶数，所以如果和的**奇偶性不对显然无解**。
- 可以发现，能把两个距离为奇数的位置一个+1 一个-1，能把两个距离为偶数的位置都+1/-1。
- 如果有奇环，因为原图连通，故任意两点都存在长度为奇数/偶数的路径，故一定有解。
- 如果没有奇环，那么是二分图，那么左部之和减右部之和不变，判断一下即可。

1270E

- 按照经典套路，**考虑奇偶性分组**。按照 x 坐标奇偶性，y 坐标奇偶性拆成四个组。
- 同一个组的距离的平方都是 4 的倍数。接下来分类讨论。
- 如果 0,0 和 1,1 至少有一个非空，0,1 和 1,0 至少有一个非空，那么 0,0∪1,1 为一组，0,1∪1,0 为一组。
- 否则如果 0,0 非空且 1,1 非空，或 0,1 非空且 1,0 非空，那么分开即可。
- 如果只有一个组非空，把横纵坐标/2 继续递归即可，显然递归层数不会超过 log。

1158B

- 考虑一种构造是 1,m 个 0,1,m 个 0,...,1,k(0<=k<=m)个 0。
- 串长 l(m+1)+1+k，最短唯一子串为(l-2)(m+1)+3+k。（形如 0,1,m 个 0,...,1,k 个 0,0/1）
- 发现总长比最短子串长 2m。m=(n-k)/2，那么总长就比最短子串长 n-k。
- 但是有个问题，如果原串的长度比 2(m+1)要小，此时最短唯一子串长度为 2。（形如 0,1）。
- 而 n<2(m+1)即 n<n-k+2，说明只有 k=1 才会出现这种情况，特判即可（形如 1,n-1 个 0）.

1196F

- 由于边权非负，只需要考虑**最短的 k 条边的子图**。
- 现在只有不超过 2k 个点，k 条边，可以直接暴力跑出每个点为起点的最短路。
- 还有一种思路是在外面维护最小值，找当前最小值最小的起点增广，可能可以少个 log。

1477C

- 如果构造的方案存在钝角∠ABC，那么根据**大边对大角**，AC 一定严格比 AB 和 BC 长。
- 所以如果 AB 比 AC 长就不存在这个问题，每次取离 A 最远的 B 即可。

1282D

- 先询问单独的 a，如果有 a 返回的是 n-1，否则返回的是 n。
- 假设有 a，得到 n，询问 n 个 a，得到的结果是其中 b 的个数。
- 如果返回的=0 说明没有 b，直接结束；如果返回的>=n 说明没有 a，直接输出。
- 再**逐位将 b 替换成 a 测试**，看答案是否变小，如果变小代表这一位是 a。

1023E

- 考虑没有只能询问曼哈顿距离>=n-1 的限制，只需要每次询问向下走和向右走可不可以走到右下角。
- 有这个限制后，由于你每次走都使曼哈顿距离减 1，你考虑对于后半部分倒过来询问，从右下走到左上。
- 关键在于你这样两部分不一定能够拼起来。所以你**前半部分尽量往下**，最后得到的使 x 最大，y 最小的方案；**后半部分尽量往左**，得到的是使 x 最大，y 最小的方案，就能拼起来了。

1583E

- 根据经典套路**先求一棵生成树**。询问给路径上的边都+1，这个和路径具体是什么有关系。
- 考虑设 f[u]表示 u 相邻的边之和的奇偶性，这个和路径具体使什么没关系，一次操作给两端点都异或 1。
- 如果最后**存在 f 是奇数显然无解**，否则叶子节点的那些边已经合法，那么推到他们父亲连到父亲的边也合法，一直推可以说明所有的边都合法，即只需要取一棵生成树，合法的依旧合法。

1311E

- 最小的方案是一棵完全二叉树，最大的方案是一条链。考虑**从最小到最大调整**。
- 先取一条从上到下的链，比如 1,2,4,8,…，然后按深度从大到小加入其它点。
- r=0 构造直接结束。否则找到那个点，如果能加到链底就加到链底，r 减小，否则加到链中间，r 变为 0。

1543D2

- 每次操作可以看作是每位模运算下的减法 z=(y-x+k)%k，它**和模运算的减法一样**满足很多限制。
- 考虑枚举答案 i，判断初始的 x 是否等于 i。比如开始 i=1，x⊖1，然后判断是不是 0。
- 如果不是，现在要判断是不是 2，可以通过(2⊖1)⊖(x⊖1)=2⊖x，3 可以通过(2⊖3)⊖(2⊖x)=x⊖3 等等。

748D

- 一定是中间一个回文串（或者没有），然后左边是若干个 s，右边是若干个 rev(s)。
- 对于代价和为正的非回文原串和反串一定选，代价和为正的回文原串和反串先选。
- 最后看看要不要拿掉选的回文串对中的较小值或加入未选回文串的最大值。

804C

- 先给 1 号点上的所有种类染色。考虑现在到点 u，这个点上还有些种类没被染过色。
- 这些没被染过色的，可以给它们分配和 u 中染过色不同的，且两两不同的颜色。
- **尽量分配编号小的颜色**，种类数应该是每个点上种类数 max，可以发现没有比这样更少的方案。

723E

- 显然度数是**奇数的永远不可能合法**。

- 如果只有度数为偶数的点，根据套路，可以考虑欧拉路，恰好有 n 次进入 n 次退出。
- 如果有部分度数为奇数的点，考虑把它们度数也变成偶数，即加一个新点连向所有度数为奇数的点。这样所有度数为偶数的点都合法。

1552E

- 记 $c=\lceil n/(k-1)\rceil$。
- 考虑先取出第二次出现位置最小的 c 个区间，将[第一次,第二次]做为它们的区间。
- 考虑从剩下的取出第三次出现位置最小的 c 个区间，将[第二次,第三次]做为它们的区间。
- 以此类推共取 k-1 次。可以发现**不同组的区间显然两两不交**，故一个位置被覆盖的次数最多为 c。

1148E

- 首先**总和不变**。可以把操作看成对于 s_j-s_i>=1（=1 相当于交换顺序，无关紧要）的两个数，s_j--，s_i++。
- 对 s_i++，s_j--，只要对值相同的取最后/最前的一个，相对顺序就不会改变，说明存在操作使**相对顺序不变**。
- 将 s,t 排序，设 a_i=t_i-s_i。现在要把前面的某个-1，后面的某个+1。可以发现有解相当于要求前缀和>0。
- 具体的构造方法类似于括号序，看栈底和当前数的大小关系即可。

576B

- 首先**树相同必须重心相同**，一棵树最多有两个重心，说明必须存在一个<=2 的循环。
- 如果存在不动点即 p_i=i，可以造一个以 i 为根的菊花。
- 否则存在对换 p_a=b,p_b=a，考虑断开 a,b 的边，那么发现两部分排列后将会变成对方。
- 所以如果存在长度为奇数的循环则无解。否则将循环上奇数位置连 a，偶数位置连 b 即可。

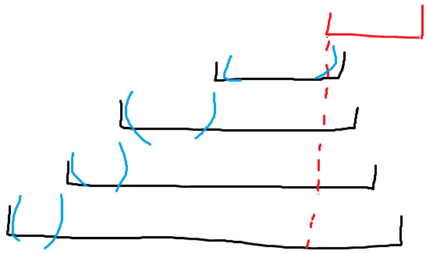
1550D

- 把式子移项变成 a_i-i=-(a_j-j)，那么这些数的**绝对值一定相同**。
- 设绝对值为 x，那么有 n/2 个为+x，n/2 个为-x。(一个取上整一个取下整)
- l-i<=a_i-i<=r-i，大概考虑一下每个 i 能取+x 的最后时间，能去-x 的最后时间，再计算一下即可。

269C

- 由于最后是 DAG，考虑一个类似于**拓扑排序**的做法。
- 首先入度为 0 的应该是 1 号点和无边相连的点。
- 那么加入队列的，没访问的邻边都是出边，更新一下，看看另一端能不能加入队列。

1278E



- 有边相当于左边相交或右边相交（但是不能同时）。
- 和儿子用左边相交，和父亲用右边相交即可。儿子之间在右边包含。

453C

- 特判度数都为偶数的情况。
- **取一棵生成树**，如果不连通只能有一个块里有度数为奇数的点，其它块不用考虑。
- 对于一个点，递归它的子树，然后看当前点的奇偶性对不对，如果不对，走到父亲再走回来就可以调整奇偶性。
- 但是这样无法调整根节点。如果根节点不对，可以把最后走回根节点的那一步删掉。

1158C

- 先考虑没有-1 的情况，存在 i,j，**$i<j<nxt_i<nxt_j$ 一定无解。否则可以构造出解**。
- 对于后缀最大值（$nxt_i=n+1$），它们之间的小区间又是独立的（$nxt_j<=i$），递归下去做。
- 具体实现可以类似于建树，i 的父亲是 nxt_i，层数低的比层数高的值大，同层编号小的比编号大的值大。
- $nxt_i=-1$ 的情况，直接取 $nxt_i=i+1$ 即可，这样不会使得原来有解的变成无解。

1285E

- 先改成左闭右开区间后离散化，现在变成：
  1. 区间+1，最后求非 0 连续段个数。
  2. 询问区间连续 1 的段的个数
- 第一个可以差分，第二个前缀和计算一下。

995A

- 开始直接能停进去的就停进去。如果此时**还是满的那就无解**。
- 否则所有车顺时针转一下再停，然后不停的转，直到所有车都停下。
- 每辆车最多转一圈才能停进去，还要花一次停进去，用的次数远小于 20000。

1468H

- 首先每次减小 k-1 个，(n-m) mod (k-1)不为 0 显然无解。
- 设 d=(k-1)/2，考虑最后一步，中心是 b_x，删去了中心左边的 d 个数，右边的 d 个数。
- 如果不存在 x 使得 b_x 左/右边有 d 个非 b 中的数，那么显然无解。
- 否则你在前面删的时候，k 个数都取要删的数，一定能调整成最后 b_x 左右各 d 个数。

549B

- 设 b_i 表示当前 i 收到的消息数量，初始 b_i=0。如果不存在 a_i=0 那么直接结束。
- 每一次找到 a_i=b_i，钦定 i 加入 party，称这个操作为 fix。
- 那么显然每个点 fix 过后 b_i>a_i，不会再次被 fix。如果某一步找不到就结束。

713B

- 先二分分界线，两个矩形一定会在 x/y 中一维是不相交的。
- 二分完分界线再二分矩形的上下左右边即可。

901B

- 没系数和值域限制的话显然直接斐波那契数列就行了。
- **系数只有 0,1 的话，正常的+/-和在 mod 2 意义下做操作次数是一样的**。
- 直接 p_i=x*p_{i-1}+p_{i-2} mod 2 即可。

550E

- 特判 n=1。如果**最后一个数是 1**，那么无论怎么操作，最后一个数的值一定是 1，**无解**。
- 如果倒数第二个是 1，那么前面怎么合并都是 1，最后和 0 合并即可。
- 000 可以变成 10，即只要最后 0 的个数=1 或>=3 都一定有解。
- 考虑最后 0 的个数为 2，即 xxx...1..100。如果之前还有 0，那么可以把 1..10 合并成 0，那么最后就有 3 个 0，有解；否则只能是 1..100，不能操作 00（最后一个数会变成 1），操作 11 或 10 都是减少 1 个 1，最后只能变成 00，00 显然没有合法方案。

811D

- 首先 BFS 出一条合法路径。沿着合法路径走。
- 如果走的结果和预期结果不一样，那么说明这个方向被恶搞了，调整一下即可。
- 这样不可能走到危险格子，因为你只有在第一行才不知道 UD 有没有换，在第一列才不知道 LR 有没有换，如果被换了你只会撞到边界，不会出问题。

1098B

- 一种是隔行相同，隔列字符集合相同；一种是隔列相同，隔行字符集合相同。(或两者都有)
- 枚举一下是第一种还是第二种（2），枚举隔行/列相同的集合和顺序（4*3），枚举另一种的顺序（2），枚举另一种隔列/行变不变（2），然后算一下即可。

584E

- 将 p 变成 1,2,...,n，**变成将 s 交换成升序的最小代价**。
- **答案的下界是|i-p_i|之和/2**，达到下界必须每次交换都没有浪费。
- 考虑从大到小确定数的位置，枚举 i 表示>i 的数都归位了，现在要将 i 归位。
- 找到 i 右边第一个 j 使得 p_j<=pos_i 的位置，交换。这样显然不存在浪费，最后也一定会归位。(没归位前右边

一定有<=pos_i 的数）

266C

- 递归，首先保证最后一列无 1，那么再交换有 1 的行和最后一行。
- 可以去掉最后一行，最后一列，再重复上面的步骤。

815B

- 找规律，n%4=2 时，第 i 项对答案的贡献是 C((n-2)/2,(i-1)/2))。
- 暴力，直到 n%4=2 即可。

1117E

- 由于 26^3=17576>15000。
- 先询问一次 aaa...aaa(26*26)bbb...bbb(26*26)ccc...ccc(26*26)...，知道在哪个大块。
- 再询问一次 aaa...aaa(26)bbb...bbb(26)ccc...ccc(26)...，知道在哪个小块。
- 最后问一次 abc...xyzabc...xyz...，知道在哪个位置。

1004D

- 首先可以钦定放的坐标 x<=(n+1)/2,y<=(m+1)/2，否则可以对称到这个范围。
- 那么离 x,y 最远的位置就是(n,m)，值为 n+m-x-y，那么 t 中数的最大值也就等于 n+m-x-y。
- 如果没有卡边界，那么距离 x,y 为 d(d>=1)的位置有 4d 个。
- 如果 x>d 且 y>d 显然不会卡边界，最小被卡边界的为 min(x,y)=d。
- 假设 x<=y，根据最大值解出 y，枚举所有位置判断即可。
- 由于 n*m=t，枚举 t 的约数作为 n,m 的值。

1283F

- 每个数出现次数为儿子个数，父亲出现的一定比儿子早，没有出现的点即为叶子节点。
- 第一个数就是根节点，接下来考虑连边情况。
- 将叶子节点加入小根堆，从后往前枚举 i，弹出堆顶 u，将 a_i 和 u 连边，如果 a_i 的儿子已经连完了就加入堆。由于每次都取最小的，所以子树权值和与根节点权值没本质区别。

1276C

- 不妨假设 R<=C。那么选进来的数出现次数都不超过 R。
- 固定了 R 之后，每种数就尽量选，看最多能选多少，再确定 C。
- 构造的方法是不停地往右下走，如果走过了就再往右。

1244F

- 先特判 BW…BW 或 WB…WB 的情况。
- 对于**长度>1 的段，显然不会改变颜色**。
- 对于交替的段，一次操作会使左端归到左边的段，右端归到右边的段，中间的翻转。具体是怎样可以通过比较段的长度和 k 的大小关系得到。

1513E

- **如果比平均值大，就一定是源，否则比平均数小只能是汇**。和平均值相同的随意（无法操作）。
- 设<的个数为 a，=的个数为 b，大于的个数为 c。
- 如果 a=1 或 c=1，无论怎样排列都可以。方案数为 $n!/(t_1!t_2!\dots t_k!)$
- 否则 a>1 且 c>1，源点必须在一起，汇点也必须在一起，否则会出现多种不同的答案。方案数为 $C(n,b)*a!/(p_1!p_2!\dots p_x!)*b!/(q_1!q_2!\dots q_y!)$。

356C

- 设为 1,2,3,4 的个数分别为 a,b,c,d。最后的目标是 a,b 清零。
- 首先一定是 12 变成 3，现在 a,b 最多有一个非零。
- 如果 a 非零尽量 111 变成 3（2 次），b 非零尽量 222 变成 33（2 次），比从 4 移出来要优。现在 a,b<3。
- 如果 a=1，c>=1 则 13 变成 4（1 次），d>=2 则 144 变成 333（2 次），否则无解。
- 如果 a=2，c>=2 则 1313 变成 44（2 次），d>=1 则 114 变成 33（2 次），否则无解。
- 如果 b=1，d>=1 则 24 变成 33（1 次），c>=2 则 233 变成 44（2 次），否则无解。
- 如果 b=2，则 22 变成 4（2 次）。
- 发现上面产生的无解情况即总和为 1,2,5 的情况，并不会把有解判成无解。

746G

- 经典套路，**从最小向最大/最大向最小调整**。
- 最小的方案是同层的点尽量选不同的父亲，最大的方案是同层点选相同的父亲。
- 调整的过程就是从下到上，把每一层的点清空到只剩一个点（清空的意思是它没有儿子）。
- 没清一个点答案都会增加 1，故如果有解一定能找得到。

925C

- 设当前异或值为 s，那么之后能加的数 v 要满足 s xor v>s，即二进制下 s 在 v 的最高位上的值为 0。
- 可以发现**最高位比较低的并不会影响最高位高的决策**，考虑从大到小加入最高位为 i 的数。
- 现在序列里已经有最高位>i 的数了，要加入最高位为 i 的数（设有 y 个），由于加入之后并不会影响那些大数的大小关系，但你新加进去的数的位置必须满足当前前缀异或起来第 i 位是 0，考虑最高位>i 的数里有多少个第 i 位为 1（设有 x 个），如果 y>x+1 则无解，否则一定可以加入。

1593G

- 考虑**哪些中括号一定要变成小括号**，可以发现奇偶性相同的位置，不能有一对匹配的括号。
- 设有 a 个奇数位置的中括号，b 个偶数位置的中括号，那么至少要|a-b|个中括号要变成小括号。
- 将相邻的奇偶性不同的一一匹配后删去，剩下的转为小括号。
- 可以发现小括号一定有合法的匹配方案（中括号区间内，把更小中括号包含的删掉，剩下的长度还为偶数）。

610D

- 离散化，将同一行相交的合并，同一列相交的合并。
- 竖着扫描线，遇到横的左端点就树状数组上+1，横的右端点就树状数组上-1，竖的就树状数组上求和。

260D

- 把黑和白的分开，每次选出黑和白最小的两个权值，在它们之间连它们最小值的边，然后删掉一个点。
- 这样显然不会产生环，权值也符合条件，但是不一定连通。
- 找到最大的那个块（大小显然>=2），里面一定有黑点和白点，将其它的块都连权值为 0 的边上去即可。

612E

- 这种看上去就是**对排列建图**。考虑把 q_i 变成 q_{q_i}会怎么样。
- 对于长度为偶数的环 1->2->3->...->2k->1，会拆成 1->3->5->...->2k-1，2->4->6->...->2k 两个长度为 k 的环。
- 对于长度为奇数的环 1->2->3->...->2k+1->1，会变成 1->3->5->...2k+1->2->4->6->...->2k。
- 那么对于 p 长度为奇数的环，换一下位置调整成 q 中的环。p 中两个相同的长度为偶数的环合并成 q 中的环。

1254C

- 和正常求凸包一样**考虑相邻两个点**。看其他点到这两个点组成直线的距离，逆时针方向上一定是先增后减的。
- 找到那个最远的点，分割成两部分。现在我们想知道哪些在逆时针到最远点之前的半边。考虑从小到大加点，维护一个栈，如果当前点和栈末尾两个点形成的角不是逆时针，如果不是，说明栈末尾的点是假的，弹出。再将当前点加入栈，这样就分成了两部分，需要 2(n-2)步。
- 现在关键在于如何找到两个相邻的点。钦定一个点 i 是 1，另一个点 j 初始是 2，然后枚举剩下的点，看有没有点在 ij 的逆时针方向，如果有就替换。

906B

- 不妨假设 n<m，有一种想法第二行是第一行+2，第三行是第二行+2...（存的是第 i 行的第几个位置）。
- 若 m>4，第一行有一种显然的**构造是 m,m-2,m-4,...，然后再 m-1,m-3,m-5,...**。m=5 为 53142，m=6 为 642531，它们无论加上什么都会保持相邻绝对值差>1。
- **小范围可以暴搜。**

772C

- 考虑当前的前缀积为 a，你要变成 b，需要填一个 x 满足 ax mod m=b，转化为 ax+my=b。
- 如果 gcd(a,m)不是 b 的约数则无解，否则存在方案。即 a 能到所有 gcd(a,m)的倍数。
- 没必要暴力连边，gcd(a,m)也只有 m 种取值，连一条 a,gcd(a,m)的 0 边，gcd(a,m)再连到它的倍数。
- 这样边数是 O(m log m)的，再用 0/1 最长路求出前缀积，还原序列可以用 exgcd 实现。

272E

- 考虑一个不合法的点 u，当且仅当它有两条或三条不合法的边。那么翻转 u，u 变得合法，**且不合法的边严格减少。**
- 由于到 0 条不合法的边的时候所有点一定都合法，故过程一定会结束。

327E

- 很容易想到的做法是直接状压 dp，2^n 记录状态，转移使用 lowbit 优化。

936C

- 你可以做到**翻转一个后缀** p..n，方法是 shift n 再 shift p-1。
- 你还可以做到将**最后一个数移到开头**，方法是 shift 1。
- 那么就好做了，考虑 i 步之后，前 i 个位置分别是 t[n-i+1],t[n-i+2],...,t[n]。
- 你现在要把 t[n-i]（在位置 p）移到开头，你先翻转后缀 p..n，再把最后一个数移到开头即可。

858F

- 依旧**对每个连通块取一棵生成树**。考虑每一个连通块。
- 连通块本身就是树，那么考虑从下往上，如果这个点的儿子的边能匹配完就匹配，不能匹配完就和父亲边匹配。这样最后最多剩下一条边，这时已达到最大方案（m 是奇数不可能都选，m 为偶数不可能有一条不选）。
- 不是树也一样，只是匹配的时候把返祖边也考虑进去，依旧是能能匹配完就匹配，不能匹配完就和父亲边匹配。

441D

- **根据经典套路，先对排列建图。**根据经典结论，f(p)=n-环数（每个环需要的次数是环长-1）。
- f(p)<m 说明要减少环的数量，选两个环上的两个元素交换一下，就把环接起来了，数量-1。
- f(p)>m 说明要增加环的数量，选一个>2 的环交换两个，数量+1 或者选一个=2 的环交换，数量+2。
- 操作的时候按照字典序最小的来即可。

962E

- 显然没必要给 BR 连边。
- 考虑 BR 会不会和**左/右方向上第一个 P 之后的位置连边**，发现肯定**不如连到第一个 P**，第一个 P 再连出去。
- 同理可以证明两个 P 中间的连边是 PRR...RP，PBB...BP 的形式。

962E

- 最后答案一定是一个区间，考虑如果求出了最小值，那么把那些 r=1 的位置的 r 变成上一个数，在把 1 单独当一个数，答案就+1 了，最大值是 1 的个数。
- 记 a_i 表示 2^i 的个数，b_i 表示(2^i,2^(i+1))中的数的个数。
- 考虑二分最小值，具体判断方法是，从小到大枚举 i，记录还能在后面继续加的数的数量 x 和之前剩余的可以当 r 的个数 y，尽量加 a_i，如果 a_i<=x，则用完 a_i，剩下的尽量用 y，x 变成 a_i；否则将 a_i 多余的给 y，y+=a_i-x。

226D

- 和 272E 类似，每次选择和为负数的一行/一列的翻转，**使矩阵总和至少+2**。
- 由于绝对值之和 mx<=1e6，且达到 mx 后一定合法，所以可以直接暴力翻。

460D

- k=1 的话选 l，r<l+4 可以暴力。
- 令 u=l+(l&1)，k>=4 可以选 u,u+1,u+2,u+3（答案为 0），k=2 可以选 u,u+1（答案为 1）。
- k=3 要么用 k=2 的构造方法，要么构造 a>b>c 异或值为 0。
- 枚举 a 的最高位 k，a,b 都会有 2^k，c 没有
- 枚举 c 的最高位 i，那么 a 只会有 c 的最高位（尽量小），剩下的位给 b，c 最优是(2^i)-1（尽量大）。

847D

- 枚举 i，表示**必须要走到位置 i** 后直接结束。拿一个堆维护之前的狗粮能吃到的最早出发时间。
- 那么堆里>=T-i 的就可以弹掉了，想要吃它们走不到 i。
- 如果当前位置本身就走不到就直接退出。否则加入当前狗粮的出发时间 T-i，判断堆的大小即可。

754C

- 对读入的字符串处理一下，这句话里出现过的用户一定没有发这条消息。
- 设 dp[i][j]表示第 i 条消息有没有可能是第 j 个用户发的。
- 考虑 j 有没有可能发消息，就看这句话有没有出现它和上一行有没有除了 j 发消息的可能性。
- 最后倒推方案即可。

670F

- 首先你可以知道长度的位数，如果长度的位数为 i 当且仅当 10^{i-1}+i<=n 且 n<=10^i-1+i。
- 最小的方案形如 t+00..+11..+...+99...或 1+00..+11..+...+t[0]..+t+...或 1+00..+11..+...+t+t[0]..+...
- 都算一下取个最小值。

12E

- 这题感觉比较离谱…由于主对角线要求是 0，由于对称性最后一行和最后一列也相等，即你把 a[i][n]（也等于 a[n][i]）填到 a[i][i]其它限制成立的条件也是一样的。
- 你现在相当于只要求是对称的和不同的两个限制，这样可以轻易的构造 a[i][j]=(i+j)%n+1。最后再令 a[i][n]=a[n][i]=a[i][i],a[i][i]=0 即可。

354E

- 对当前这一位，6 个数字中哪些是 0，哪些是 4，哪些是 7 的顺序无关紧要，只和每种的个数有关。
- 你知道了进位就知道这一位可能能填的数，填就是了。

429C

- **每个点的儿子个数为 0 或 2 的图**有个重要的性质：**设叶子节点个数为 x，那么 1+2(n-x)=n，x=(n+1)/2**。
- 儿子个数可以>2 的话叶子节点数可能更多。由于非叶子节点两两等价，设 f(s)表示 s 集合的点是否能构成一棵树（非叶子节点数为|s|，总点数为 s 中 c_i 的最大值）。
- 转移的时候需要用 g(s,i)表示选了 s 的点，为若干个森林（至少 2 个），叶子节点个数为 i 是否可行。
- 注意特判叶子节点数>12 和 n=1。

475C

- **注意到每次只能向下或向右，所以路径长度是 O(n+m)的**。
- 找到最靠上其次最靠左的 X，它横向的连续 X 有 x 个，纵向的连续 X 有 y 个。
- 那么考虑第一步如果往右说明纵向的必须为 y 个，往下说明横向的必须为 x 个。
- 可以直接暴力枚举另一个方向的长度，然后暴力判断是否合法。复杂度为 O((n+m)^2)。

1039A

- 题目给定 a_i 是单调递增的，那么 a_i+t 也是单调递增的，且 b_i 是单调递增的，所以 b_i>=a_i+t。
- **那么说明 x_i>=i，否则一定无解**。否则若 x_i=c，那么删除 b_c 和 a_i 后对应位置依旧需满足 b>=a+t。即变成 b_i>=a_{i+1}+t,b_{i+1}>=a_{i+2}+t,…,b_{c-1}>=a_c+t。**满足条件的情况下要求 b 尽量小**（否则 x_i 可能>c）。
- 最后再判断一下 x_i 是否=c 即可。

277B

- 构造一个上凸壳，构造一个下凸壳，两个凸壳放远一点。
- 一个凸壳选了 3 个点后，另一个凸壳不能选 2 个及以上的点。
- 构造上凸壳为(x,x^3)，下凸壳为(x,-x^3)，为了防止出现三点共线，给 y 加上 1e7。
- 可能会出现上面选两个下面选两个的情况，所以 m=3 可能会出问题。
- 分析发现 m=3 时，n>=5 无解。（你考虑拿出最靠外的三个点，固定三角形内一个点 D，发现 E 无论放哪个区域都会使得有 4 个点的凸包）
- n=3 和 n=4 上述构造不会出现问题。

196C

- 对点进行极角排序，选最左一个点当根。然后，对于大小为 x 的儿子，再**选出极角序最小的 x 个**。
- 递归即可。

509D

- 这么经典的题为什么过的人这么少。。。
- 考虑**如果得到了一组解 a,b，那么 a-x,b+x 也是一组解**。
- 那么**不妨假设 a[1]=0**，那么 b[i]=a[1][i]，同理可以解出 a[i]=v[i][1]-b[i]。
- 现在要所有(a[i]+b[j]-v[i][j])%k=0，对所有|a[i]+b[j]-v[i][j]|求 gcd。
- 如果 gcd<=矩阵 max，那么无解，否则取 k=gcd 即可（注意特判 gcd=0）。

1157G

- 如果 n=1 直接 Yes。
- 现在要变得有序，考虑枚举一行，这一行之前都是 0，这一行之后都是 1。
- 对每一行预处理一下，第一行的两种方案是否使这一行变成全 0/全 1/先 0 后 1。
- 如果这不是第一行，那么第一行全为 0 的方案只有 2 种（对应行翻不翻），前缀和判断一下。
- 如果这是第一行，那么最后一行全为 0 的方案只有 2 种（对应行翻不翻），直接暴力。
- 时间复杂度 O(nm)。

350E

- 根据 floyd 的原理，你枚举的标记点实际是路径经过的中间点（不包括端点）。
- 对于任意 u,v，假如不存在一条最短路，使得路径上全是标记过的节点，那么它们的最短路。
- 考虑构造，先选出一个标记点 a，把所有非标记点都连到 a，其它的标记点连向除了 a 外的任意点。
- 如果边数不满 m 的话，可以任意连边（前提是不能连到 a）。
- 无解的情况是没有标记点（k=n）或 m>C(n-1,2)+n-k（边连完后 m 还有剩余）。

97B

- 如果没有点数限制就可以把整个矩阵填满。
- 有点数限制可以考虑把所有点按照先 x 后 y 排序，分治，考虑中间点 mid（x[mid],y[mid]）。
- 两边的点**向中间点所在的 x=x[mid]直线做投影**，那么<mid 的点和>mid 的点两两都满足条件，直线上的点和其它点（和以后可能加入的点）也都满足条件，递归到[l,mid-1]和[mid+1,r]即可。

141E

- 如果 n 是偶数，那么 n-1 是奇数，显然无解。
- R 边最多的方式就是 R 边边权为 0，B 边边权为 1 的 MST。（即先加 R 边）
- B 边最多的方式就是 B 边边权为 0，R 边边权为 1 的 MST。（即先加 B 边）
- 如果 R/B 边最多也达不到一半，那么也无解。
- 否则考虑**从 R 边最多开始调整**，每次加入一条不在里面的 B 边，找环上的一条 R 边删除即可（如果找不到就

不加)，直到 R 边只剩一半就结束。

81D

- 首先每个数放的次数不能超过[n/2]，所以 a_i 可以和[n/2]取 min。如果 a_i 之和不到 n 那么无解。
- 如果有一种数的数量是[n/2]，考虑 n 的奇偶性，如果是偶数那么在所有偶数位置放这个数，奇数位置任意放其它数；如果是奇数依旧在所有偶数位置放这个数，剩下必然至少有两种数（否则总和不到)，各取一个放 1,n 两位置，剩下随便放。
- 否则，对于每种数，按照顺序 1,3,5,7,...,2,4,6,8,...放下去即可。

432E

- 刚开始想的是每次找 Up-Left most 的位置，填尽量小的数，然后边长尽量长。
- 后来发现不对，你填尽量小的数，可能别的数更小但你不能填。
- 应该改为填尽量小的数，**填到一个这一行能填更小字母的位置或不能填的位置**。

132D

- 你考虑长为 len 的一段数的表示方法：
  1. 给每个 1 的位置+；
  2. （在最后一个位置为 1 的情况下）+2^len，-2^0，再给每个为 0 的位置-。
- 有一个容易想到的错误贪心：对于每一段 1，如果长度为 1 就直接+，否则在最低位-，最高位+1 的位置+。
- 很显然第二种表示方法只有第一个位置和最后一个位置都为 1 才有意义，否则你可以删去前后的 0。
- 继续考虑，发现第二种如果其中有连续的>=2 个 0，你可以把它拆分成前后都用第二种不会更劣。
- 对于长度 len>=2 的，开头结尾没有 0 的，其中没有连续>=2 个 0 的，且倒数第二个位置是 1，用第二种不比第一种更劣（1 的个数>=0 的个数+2)，也不比拆分更劣。
- 在开头加两个 0，从低位往高位考虑，如果这一位和下一位都是 1，则这一位为第二种的起点，直到出现某一位和它下一位都是 0 结束，否则的话用第一种。

748F

- 如果你能取一个根，使得每棵子树的关键点数<=k，那么**答案为 1**。
- 这玩意可以和求重心一样求，这个点一定存在。你考虑和从根节点开始走关键点数最多的儿子。一定能走到点满足条件。

370D

- 一定有一维卡边界，你可以知道正方形的边长是 max(xmax-xmin,ymax-ymin)。
- 枚举卡边界的是 x 维还是 y 维，已知边长，看另一维有没有合法方案即可。

1231E

- 排序后不同显然无解。

- 考虑被保留下来的，相当于你要找一个 S 的子序列，T 的子串，使长度尽量长。
- 枚举 T 中的左端点，枚举 S，看最多能有多长。

424D

- 枚举上边界和下边界，从左到右枚举右边界，维护所有左边界加的权值。（为左列上的贡献-上下行贡献前缀和）
- 每行贡献前缀和可以开始就算出，左列的贡献也可通过前缀和算出。
- 计算答案的话在 set 找离期望值最近的两个（小/大），判断一下。

313E

- 加起来的和 mod m 只有<m，不用减 和 >=m，要减去 m 两种可能。
- 开个桶记录每个数字出现次数，你显然希望 mod m 和为 m-1 的尽量多，那么只有 b=m-1-a(a+b=m-1)的情况（不存在 a+b=2m-1）。
- 从小到大枚举 a，如果对于一个 a_i 找不到多余的 b=m-1-a 的话，就先压进栈里，如果之后的 b=m-1-a’多余的话可以和栈里的匹配（b 和更大的 a 匹配然后-m 一定不优，a 和之后更小的 b 匹配也一定不优）。
- 对于剩下来栈里的 a，和多余的 b 之和一定>m，从大到小枚举 b，弹栈（一定-m 那肯定越大越好）。
- 最后排序输出即可。

271E

- 这道题是不是也属于比较阴间的找不变量？**考虑所有卡片 y-x 的差值。**
- 1 操作出现一张差值不变的卡，2 操作出现差值/2 的卡，3 操作出现两个差值相加的卡。
- 设 d=y-x，那么不停/2，得到 d’(奇数),d=d’*2^k，那么无论哪一步产生的都是 d’的倍数。
- 猜测只要 d’|gcd(a_1-1,a_2-1,…,a_n-1)就可以了，证明:
- 定义 13 操作：(x,y)通过一操作可以获得(y,2y-x)，再通过三操作获得(x,2y-x).
    1. 必要性：你每次只能得到 d’的倍数，如果 gcd 不是 d’的倍数，说明有数不是 d’的倍数，凑不出来。
    2. 充分性：开始是(x, x+d’*2^k)，通过一次 13 操作可以变成(x, x+d’*2^{k+1})，通过若干次能变成(x, x+d’*2^{100})，再利用 1 操作变成(2^{100}, (1+d’)*2^{100})，再不断使用 2 操作变成(1,1+d’)，之后可以通过 13 操作变成(1,1+2d’),(1,1+3d’),…，由于 d’|a_i-1，所以一定能表出所有数。
- 枚举 gcd 的奇约数，枚举它的 2^k 倍即可得到 d，即 y-x 之差，计算方案数即可。

134C

- 拿个堆维护每个人手上还剩的自己的牌的数量。考虑自己的牌最多的人 x，尽量要让自己的牌最多的人牌数最少和剩下的还有牌的人最多，就和剩下的自己的牌的数量从大到小前 x 大的人交换，使它们自己的牌都-1。

42C

- 不断重复，找到最大值，如果等于 1 直接结束。
- 如果是偶数：
    1. 左边的是偶数，两个都/2，**总和一定减小**
    2. 右边的是偶数，两个都/2，**总和一定减小**

3. 否则两个都是奇数，先给左边和自己+1，自己和右边+1
   - 如果原来最大值不是 4 的倍数，左边和自己/2，自己和右边/2，**总和一定减小**
     (a,b,c)->(a+1,b+2,c+1)->((a+1)/2,(b+2)/4,(c+1)/2)
     (a+1)/2<=a,(b+2)/4<=b-1,(c+1)/2<=c，总和不超过 a+b+c-1
   - 如果原本最大值是 4 的倍数，左边和自己/2，左边和自己+1，自己和右边/2，**总和一定减小**
     (a,b,c) ->(a+1,b+2,c+1)->((a+1)/2,(b+2)/2,c+1)->((a+1)/2+1,b/2+2,c+1)->((a+1)/2+1,b/4+1,(c+1)/2)
     (a+1)/2+1<=a+1,b/4+1<=b-2,(c+1)/2<=c，总和不超过 a+b+c-1

- 如果是奇数：
  1. 左边的是奇数，两个都+1,/2，**总和一定减小**
  2. 右边的是偶数，两个都+1,/2，**总和一定减小**
  3. 否则两个都是偶数，给左边和自己+1，自己和右边/2，**总和一定减小**
     (a,b,c)->(a+1,b+1,c)->(a+1,(b+1)/2,c/2)
     a+1<=a+1,(b+1)/2<=b-1,c/2<=c-1，总和不超过 a+b+c-1

86B

- 从上到下，从左到右，先用 1*2 和 2*1 的来覆盖。剩下的空位找四连通的已经被覆盖的接在一起。
- 如果没有说明四连通都是障碍物，无解。可以证明每个块大小不超过 5。

802H

- 考虑递归解决。你现在要求答案为 n 的 s,t。n 是偶数比较好解决，递归到 n/2，n 是奇数比较难。
- 不妨换种考虑方法，求 r,t，**令 s=t+r**，使得答案为 n。
- 考虑两种变换，x 是原先没出现过的字符。
  1. （t,r）->（t+”x”,r+”xx”），那么原先的所有方案都可以在后面取两个 x 中的一个，前面的 t+”x”本身也是一种方案，现在的答案变成 n*2+1
  2. （t,r）->（t+”x”,”x”+r+”xx”），那么原先的所有方案都可以在后面取两个 x 中的一个，前面的 t+”x”有两种方案，现在的答案变成 n*2+2
- 故 n 是奇数，递归到(n-1)/2；n 是偶数，递归到(n-2)/2。

42D

- 由于路径一定是经过每个点一次，**那么设边的权值为 a[u]+a[v]，那么路径权值就固定**为 2*suma。
- 现在要给每个点分配这样的 a，使得所有边权值不同。
- 不妨考虑依次加入 1,2,...,n，那么不存在 a[u]+a[v]=a[i]+a[j](u≠v)，将之前所有 a[u]+a[v]-a[j]加入 set，这些不能选，然后尽量选最小的不在 set 中的正整数即可。
- 然后发现在 n=20 的时候 a[20]=413，符合条件。

125D

- 考虑前三个位置，必有两个在同一等差数列，枚举一下，就知道等差数列的前两项和公差。
- 由于数字互不相同，直接往后取即可。取完了之后，剩下的给第二个数列。
- 但是第一个数列可能没有这么长，可能删去了一段后缀。
- 先特判第一个数列不删，删去最后一个位置，删去最后两个位置可不可行。
- 否则至少删去三个位置，第二个数列的最后两个数固定了，可以倒推哪些数要在第二个数列，反过来判断第一

个数列即可。

323B

- n=3 的答案显然是个三角形的环，n=4 样例告诉我们无解。
- 可以发现三角形非常好，考虑每次加入两个点，这两个点和之前的所有点都形成三角形。
- 比如加入 n+1,n+2，就让 n+1->x, x->n+2, n+2->n+1，这样依旧合法。现在我们会**从 n 到 n+2**。
- 发现 n=6 也可以构造，思路是先搞两个三角形，然后再连边。一个点往另外一边的点连的边，能使得它到达另外一边的两个点，剩下的一个点，要求是它向同一边点连的边连向的另一边的点。
- n=3 和 n=6 都构造出来了，那么所有 n>=3 且 n≠4 的都可以构造，n=4 无解。

209B

- 先排序使得 a<=b<=c。在没结束时，你要让一个数至少为 2 的数-2，可以用两次（设自己为 x，另外某个非 0 数为 y，剩下的为 z）：x-1,y-1,z+1; x-1,y+1,z-1。
- 如果 a,b 奇偶性相同，至少操作 b 次。先 b-2 直到 a=b，然后再 a-1,b-1,c+1 直到 a,b 都为 0，操作次数为 b。
- 如果 a,b 奇偶性不同，那么无论怎么操作 a,b 的奇偶性都不同，那么只能 c 最后为 0，至少操作 c 次。先 c-2 直到 a=c 或 b=c，然后 a-1,b+1,c-1 或 a+1,b-1,c-1 直到 a,b 或 a,c 都为 0，操作次数为 c。

593C

- **考虑类似 CRT 一样，你构造若干个函数 f_i(z)，在 z=i 的时候返回 x_i，在 z≠i 的时候返回 0，把它们加起来**就可以得到 f(z)了，对于 g 同理。
- 考虑找到一个函数[z=0]，这个可以通过(2-||z-1|-|z+1||)/2 表示，但是题目没有除法，我们只能有 2[z=0]。
- 那么 2[z=i]*(x_i/2)也一样，即 2[z-i=0] *(x_i/2)，可以通过(2-||z-i-1|-|z-i+1||)*(x_i/2)表示，你可以表示出[z=i]x_i（偶数）或[z=i](x_i-1)（奇数），因为除以 2 会取下整。
- 不过问题不大，因为 r>=2，即使圆心是(x,y)，(x-1,y),(x,y-1),(x-1,y-1)也一定在圆内。直接这么做就可以了。

774H

- 一个长度为 x 的极长同色连续段，它对 c 数组的贡献是 c[x]+1,c[x-1]+2,c[x-2]+3,...
- 相当于给 c[x]+1，最后做两次后缀和。那么做两次后缀差分就可以得到每种极长连续段的个数。
- 然后随便怎么做都可以吧，你只要保证相邻的连续段颜色不同即可。

952G

- 这个也太憨憨了，你首先要知道+是+1，-是-1，.是输出，还得知道它存的是 signed char，可以溢出。
- 那么你就不停的-（比较好构造），然后相同就.，减号输出..\nX.\n，点号输出 X.\n 就行了。

306D

- 考虑没有边长不同的限制，那么可以直接构造正 n 边形，即每次加上(cos(2π/n*k),sin(2π/n*k))*len。

- 但是现在有限制，那么每次要给 len 加上一个比较小的值，比如 0.01，你可以确定 n-1 个点。
- 最后一个点要单独考虑，否则可能最后一个点形成的角度不对。最后一个点的 y 必须为 0，算一下长度即可。

316F3

- 这题好离谱啊。
- 先进行“收缩”，即如果一个点的周围有黑色就将它变成黑的，重复若干次。
- 然后再“展开”，即一个点周围有白色的就把它变成白的，重复若干次
- 用原图减去这张图，得到那些光线。dfs 一下即可。