A. Jujubesister

对每个值 x 统计 x 作为 a_i, a_k 时对每个询问的贡献。

查询在 x 的每两次相邻出现位置之间 $< x$ 的元素个数（升序扫 x 变为点加区间和）得到序列 p，对每个询问计算贡献：

对于出现次数超过 B 的值 x，按 x 的出现位置将询问端点离散化，贡献是序列 p 经过以下标为自变量，参数依赖询问端点的二次函数的映射后的区间和，可以预处理前缀和；

对于出现次数不超过 B 的值 x，对 (l, r) 平面的贡献是不超过出现次数平方次矩形加，可以用扫描线处理。

当 $n = m$ 时，时间复杂度为 $O(n^{\frac{3}{2}})$ 。

B. Circle of Mistery

有点烦人的题目，涉及到数据结构、贪心以及一定的性质推导。好在结论非常好猜。题解涉及到的结论的证明会在题解末尾给出。

如果 k 是负数或者 0 那么显然一眼丁真，答案就是 0 或者无解。接下来只考虑 k 是正数的情况。

首先如果权值全都是正数那该怎么做。显然我们最终构造出的排列一定恰好只有一个大环和若干个大小为 1 的自环，这个大环即是我们要找的权值和大于等于 k 的环。那么一个猜测是如果我钦定区间 $[l, r]$ 里的所有元素需要被包含进大环中，那么构造一个逆序对个数最小的大环方式即为 $\{1, 2, \ldots, l+1, l+2, \ldots, r, l, \ldots, n-1, n\}$ 这样构造。它的花费正好是区间长度减 1。显然从这个大区间中扣掉一些元素反而会增大花费，所以选择一个整区间一定最优。使用双指针即可。

然后我们发现会有负数权值，所以选择一个区间之后可能会舍弃一些点，这也意味着我们选择的环会构成一个集合。当然，我们如法炮制上面的构造，即假设我们选择了有序数组 p 这些位置的共 x 个元素，那么就能构造环 $\{p_2, p_3, \ldots, p_x, p_1\}$。由于构造的环显然要跨过每个被舍弃的点两次，所以这样的构造仍然达到了理论下界。可以发现每个被舍弃的点会额外贡献一次。虽然我们是不能舍弃最左和最右的点的，但因为我们如果进行了这样的操作那么一定能找到一个严格不劣的合法操作，所以我们默认这样的舍弃也是可行的。

有了理论基础就可以借助扫描线和贪心来解决这个问题了。如果固定区间，那么每次只要舍弃权值最小的点即可。但是这样做复杂度过高。枚举右端点并枚举所选择的后缀，由于区间的延长会使得区间长度减一这条贡献增大，所以我们舍弃的点的数量一定会严格减小。我们找到第一个可能满足条件的后缀（即正数和大于等于 k）并用贪心求出至少要舍弃多少点，接下来不断加入元素并查验是否能舍弃更少的点使得区间满足条件。在这一过程中使用优先队列维护所有可能被舍弃的点即可达到 $O(n^2 \log n)$ 的复杂度。

结论证明：使一个长度为 n 的排列形成一个大环所需要花费的最小逆序对数是 $n-1$。记 $f(x)$ 表示长度为 x 的排列形成一个大环所需要花费的最小逆序对数，我们可以归纳证明。显然 $f(1) = 0, f(2) = 1, f(3) = 2$。对于 $n \geq 4$，我们假设 $f(x) = x - 1$ 对于 $x < n$ 成立。假设 1 在排列中所处的位置为 i（注意到 i 不可能等于 1），那么其至少为排列贡献了 $i-1$ 数量的逆序对。将 1 删除后，排列会形成一个大小至少为 $n-i+1$ 的大环（读者自证不难），所以 $f(n) \geq f(n-i+1) + i - 1$，即 $f(n) \geq n - 1$，由于构造一组 $f(n) = n - 1$ 的答案是简单的，所以 $f(n) = n - 1$ 得证。

C. Cheeeeeen the Cute Cat

把一组匹配 $(x, y+n)$ 视为连边 (x, y)，那么这张图存在完美匹配即为可以将其所有点划分进若干回路中。根据竞赛图的性质，其一定存在一条哈密顿路径，所以显然最大匹配至少为 $n-1$。再根据竞赛图的性质，每个强连通分量必然存在一条哈密顿回路，所以答案为 n 等价于图中所有强连通分量大小均大于等于 2。利用兰道定理可以通过度数序列找到竞赛图的所有强连通分量。时间复杂度 $O(n^2)$，空间复杂度 $O(n)$。

D. Cirno's Perfect Equation Class

本场比赛的签到之一。虽然式子看起来很吓唬人，但是容易发现 b 是 c 的约数，所以合法的 b 数量不会太多。对 c 进行因数分解，然后枚举所有合法的 b 并解出 a，然后挨个检验合法性即可。

E. Red and Blue and Green

区间满足的性质即若相交则包含。很显然，和题面所说的一样，区间会形成树状结构。一个很显然的事实是，如果存在一条形如要求 $[i, i]$ 的区间逆序对奇偶性为 1 的限制，那么无解。为了方便实现，我们加入所有形如 $[i, i]$ 奇偶性为 0 的限制。接下来我们可以递归构造一个合法的 p：

首先令 $p_i = i$，接下来对树进行 dfs。可以发现当我们递归至某区间 $[l, r]$ 并尝试解决它时，我们已经解决了其所有儿子的限制。如果它的限制直接被满足了，那么皆大欢喜。否则，我们选择其两个儿子区间 $[a, b]$ 与 $[b+1, c]$（由于我们的处理，显然一定能找到这样的两个区间），并将 $[a, b]$ 中的 b 和 $[b+1, c]$ 中的 $b+1$ 交换位置。很显然，这样的一次交换后，其所有子孙区间的逆序对奇偶性不会改变，其自身的奇偶性恰好改变。

可以发现在上述的操作中我们始终保持了待解决区间的儿子区间 $[l, r]$ 一定恰好是 $[l, r]$ 的一个排列，所以上述操作一定能递归地找到一组解。时间复杂度 $O(n+m)$，但是为了方便 checker 实现所以没有 hit 这个上界。

F. NoCruelty

区间排序操作可以用平衡树维护每次排序产生的段，每段用 Trie 维护排序后的结果，每次修改均摊地进行 Trie 分裂合并。

查询前缀颜色数，只需维护每种颜色的最左出现位置。在 Trie 上维护每个权值的出现次数和是否是最左出现，分裂/合并时只有被分到两侧/被合并的权值需要修改出现次数和是否最左出现。查询时需要查完整的段上最左出现的值个数的前缀和，以及在查询切开的段的 Trie 上查前缀。

G. Go to Play Maimai DX

本场比赛的签到之一。可以发现 1 代表大水，2 代表手套，3 代表耳机，4 代表币。Ran 需要拿够 k 个币并且带全装备才能出勤。这题做法很多，可以使用双指针求解，也可以对每个左端点预处理其最近的右端点满足 $1/2/3/4$ 的要求然后取 $\max$。选择你喜欢的方式实现即可。

H. Nazrin the Greeeeeedy Mouse

首先考虑到 $sz_i \geq sz_{i-1}$，所以当 $m > n$ 时，只用保留最后 n 次即可。接下来发现，每次操作一定是拿走或破坏一个前缀的奶酪，所以如果我们知道了一个区间用大小为所少的包至多可以获得多少价值的奶酪的话，可以写出 dp：$f_{i,j}$ 表示当前已经拿走/破坏到了第 i 个奶酪，并且进行到了第 j 个操作，转移是简单的。

然后我们发现预处理可以使用区间 dp：$g_{i,j,k}$ 表示区间 $[i,j]$ 用大小为 k 的包能获得的最大价值，转移就相当于背包新加入一个物品。总时间复杂度为 $O(\max\{sz_i\}n^2+n^3)$，可以轻松通过。

I. The Yakumo Famliy

考虑枚举中间的区间（即 $[l_2,r_2]$），即它的异或和为 w，把左边与右边的区间抽象成两个数组 x 与 y，贡献即形如 $w\sum x_iy_j$，这显然又可以拆成 $w(\sum x_i)(\sum y_i)$。

对于左端点 i，预处理出其左侧的所有子区间异或和的和 a_i。对右侧也做同样操作预处理出 b_i。这一步可以通过拆位技巧达到 $O(n\log w)$。

考虑枚举中间的区间，贡献形如 $\sum XOR(l,r)a_lb_r$。不妨扫左端点 l，那么提出 a_l，右半部分可以通过拆位技巧维护，时间复杂度为 $O(n\log w)$。

J. OIL

题意是每次将序列分为左右两部分，将左边翻转，然后查询左右都出现过的颜色数。

可以转化为对出现次数为 2,3 的颜色分别查询满足 $i\le x<j,\ a_i=a_j$ 的 (i,j) 的数量，贡献分别为 $1,\frac{1}{2}$。 这样的 (i,j) 只有 $O(n)$ 个，且在操作前后不会减少只会改变位置。

离线处理，按 $\sqrt{n}$ 的块大小对操作序列分块。维护 (i,j) 时只需精确到 i,j 所在的块编号。

块内可以分治处理，维护按当前分治区间内的询问的 x 分块时，区间内的操作对块的重排和翻转方式（$T(k)=2T(\frac{k}{2})+O(k)=O(k\log k)$），以及两两块间的 (i,j) 的个数（$T(k)=2T(\frac{k}{2})+O(k^2)=O(k^2)$）。每块处理完后计算块内操作完成后的 a 序列。

时间复杂度 $O((n+m)\sqrt{n})$。