五月多校联合训练（HZNU 站）

题目解析

Hangzhou Normal U ACM/ICPC Team

2023 年 05 月 27 日

1 Solution

Problem A 括号

- 令 b 是数组 a 的非递减数组

Problem A 括号

- 令 b 是数组 a 的非递减数组
- 若 n 为偶数并且 $b_{\frac{n}{2}} \neq b_{\frac{n}{2}+1}$，只有满足 $b_{\frac{n}{2}} < k < b_{\frac{n}{2}+1}$ 时，左右括号数量相同。对于满足条件的 k 的括号序列都相同，所以只要选择满足该条件的任意一个 k 检验括号序列是否合法即可。

Problem A 括号

- 令 b 是数组 a 的非递减数组
- 若 n 为偶数并且 $b_{\frac{n}{2}} \neq b_{\frac{n}{2}+1}$，只有满足 $b_{\frac{n}{2}} < k < b_{\frac{n}{2}+1}$ 时，左右括号数量相同。对于满足条件的 k 的括号序列都相同，所以只要选择满足该条件的任意一个 k 检验括号序列是否合法即可。
- 若 n 为偶数并且 $b_{\frac{n}{2}} = b_{\frac{n}{2}+1}$，仅当 $k = b_{\frac{n}{2}}$ 时，左右括号数量相同。此时检验该 k 的括号序列是否合法即可。

Problem A 括号

- 令 b 是数组 a 的非递减数组
- 若 n 为偶数并且 $b_{\frac{n}{2}} \neq b_{\frac{n}{2}+1}$，只有满足 $b_{\frac{n}{2}} < k < b_{\frac{n}{2}+1}$ 时，左右括号数量相同。对于满足条件的 k 的括号序列都相同，所以只要选择满足该条件的任意一个 k 检验括号序列是否合法即可。
- 若 n 为偶数并且 $b_{\frac{n}{2}} = b_{\frac{n}{2}+1}$, 仅当 $k = b_{\frac{n}{2}}$ 时，左右括号数量相同。此时检验该 k 的括号序列是否合法即可。
- 若 n 为奇数，仅当 $k = b_{\frac{n}{2}+1}$ 时，左右括号数量相同。此时检验该 k 的括号序列是否合法即可。

Problem B 操作

- $+1$ 和 -1 操作不改变元素的相对大小，而反转操作时改变所有元素的相对大小关系，所以反转操作至多使用一次。

Problem B 操作

- $+1$ 和 -1 操作不改变元素的相对大小，而反转操作时改变所有元素的相对大小关系，所以反转操作至多使用一次。
- 将 a 和 b 数组重排为非递减数组，只对于任意的 $2 \le i \le n$ 满足 $a_i - b_i = a_1 - b_1$ 时才能将 a 数组变为 b。

Problem B 操作

- $+1$ 和 -1 操作不改变元素的相对大小，而反转操作时改变所有元素的相对大小关系，所以反转操作至多使用一次。
- 将 a 和 b 数组重排为非递减数组，只对于任意的 $2 \leq i \leq n$ 满足 $a_i - b_i = a_1 - b_1$ 时才能将 a 数组变为 b。
- 最小操作次数就是 $min\{abs(a_1 - b_1), abs(-a_1 - b_1) + 1\}$ 。

Problem C 买火柴的小男孩

- 考虑已经构造了一个 n 根火柴棒的式子。发现只需要在等号两边各找一个数字分别增加一个高位 1，此时的 $n+4$ 根火柴棒的式子依旧满足条件。

Problem C 买火柴的小男孩

- 考虑已经构造了一个 n 根火柴棒的式子。发现只需要在等号两边各找一个数字分别增加一个高位 1，此时的 $n+4$ 根火柴棒的式子依旧满足条件。
- 只要手动构造 $\mod 4$ 不同数的基础等式即可。

Problem C 买火柴的小男孩

- 考虑已经构造了一个 n 根火柴棒的式子。发现只需要在等号两边各找一个数字分别增加一个高位 1，此时的 $n+4$ 根火柴棒的式子依旧满足条件。
- 只要手动构造 $\mod 4$ 不同数的基础等式即可。
- 手玩或者打表可知下限为 12。

Problem D tententen 选月饼

- 方法一：

Problem D tententen 选月饼

- 方法一：
- 离线询问，将询问放在右端点。依次加入每个数，考虑以每个数作为右端点的合法左端点的取值。由于单调性，发现一定是一段后缀。

Problem D tententen 选月饼

- 方法一：
- 离线询问，将询问放在右端点。依次加入每个数，考虑以每个数作为右端点的合法左端点的取值。由于单调性，发现一定是一段后缀。
- 二分第一个合法位置后区间加，对于每个询问求区间和。

Problem D tententen 选月饼

- 方法一：
- 离线询问，将询问放在右端点。依次加入每个数，考虑以每个数作为右端点的合法左端点的取值。由于单调性，发现一定是一段后缀。
- 二分第一个合法位置后区间加，对于每个询问求区间和。
- 二分时的检验可以使用 ST 表。

Problem D tententen 选月饼

- 方法二：

Problem D tententen 选月饼

- 方法二：
- 固定 l，则 r 具有单调性，也就是说若 $[l, r_i]$ 不满足条件则 $[l, r_j], j > i$ 也一定不满足条件。

Problem D tententen 选月饼

- 方法二：
- 固定 l，则 r 具有单调性，也就是说若 $[l, r_i]$ 不满足条件则 $[l, r_j], j > i$ 也一定不满足条件。
- 那么我们可以通过双指针 $+map/multiset$ 或者二分 $+ST$ 表处理出每个 l 所对应的最大的满足条件的 r，并用前缀和处理出 $[1, i]$ 的答案，记作 pre_i 。

Problem D tententen 选月饼

- 方法二:
- 固定 l, 则 r 具有单调性, 也就是说若 $[l, r_i]$ 不满足条件则 $[l, r_j], j > i$ 也一定不满足条件。
- 那么我们可以通过双指针 $+map/multiset$ 或者二分 $+ST$ 表处理出每个 l 所对应的最大的满足条件的 r, 并用前缀和处理出 $[1, i]$ 的答案, 记作 pre_i 。
- 对于一次询问 L,R
  若 R 所对应的最大的满足条件的 r 就是 R, 则答案就是 $pre_R - pre_{l-1}$。

Problem D tententen 选月饼

- 方法二：
- 固定 l，则 r 具有单调性，也就是说若 $[l, r_i]$ 不满足条件则 $[l, r_j], j > i$ 也一定不满足条件。
- 那么我们可以通过双指针 $+map/multiset$ 或者二分 $+ST$ 表处理出每个 l 所对应的最大的满足条件的 r，并用前缀和处理出 $[1, i]$ 的答案，记作 pre_i 。
- 对于一次询问 L,R
  若 R 所对应的最大的满足条件的 r 就是 R，则答案就是 $pre_R - pre_{l-1}$。
- 若 L 所对应的最大的满足条件的 r 大于 R，那么 $[L, R]$ 的所有子区间均满足条件，答案为 $\frac{(R-L+1)*1+R}{2}$。

Problem D tententen 选月饼

- 不然，则存在一个位置 mid，使得 $[L, mid]$ 所对应的最大的满足条件的 r 均小于等于 R，$[mid+1, R]$ 所对应的最大的满足条件的 r 大于 R，mid 可以通过二分找到，此时答案为 $pre_{mid} - pre_{L-1} + \frac{(R-mid)*(L+R-mid)}{2}$ 。

Problem D tententen 选月饼

- 不然，则存在一个位置 mid，使得 $[L, mid]$ 所对应的最大的满足条件的 r 均小于等于 R，$[mid+1, R]$ 所对应的最大的满足条件的 r 大于 R，mid 可以通过二分找到，此时答案为 $pre_{mid} - pre_{L-1} + \frac{(R-mid)*(L+R-mid)}{2}$。
- 本质都是一样的。

Problem E 企鹅取暖

- 方法一：将询问离线。

Problem E 企鹅取暖

- 方法一：将询问离线。
- 对于操作一，建立新点，其儿子分别是 u 和 v 所在树的根

Problem E 企鹅取暖

- 方法一：将询问离线。
- 对于操作一，建立新点，其儿子分别是 u 和 v 所在树的根
- 对于操作二，建立新点，其儿子为 u 所在树的根。

Problem E 企鹅取暖

- 方法一：将询问离线。
- 对于操作一，建立新点，其儿子分别是 u 和 v 所在树的根
- 对于操作二，建立新点，其儿子为 u 所在树的根。
- 对于操作三，建立新点，其儿子为 u 所在树的根。

Problem E 企鹅取暖

- 方法一：将询问离线。
- 对于操作一，建立新点，其儿子分别是 u 和 v 所在树的根
- 对于操作二，建立新点，其儿子为 u 所在树的根。
- 对于操作三，建立新点，其儿子为 u 所在树的根。
- 每个点上携带操作相关信息以及操作的时间戳。

Problem E 企鹅取暖

- 方法一：将询问离线。
- 对于操作一，建立新点，其儿子分别是 u 和 v 所在树的根
- 对于操作二，建立新点，其儿子为 u 所在树的根。
- 对于操作三，建立新点，其儿子为 u 所在树的根。
- 每个点上携带操作相关信息以及操作的时间戳。
- *dfs* 整棵树，发现每次遇到一个点就是一个子树叶子加，其权值由子树叶子权值和以及父子结点的时间戳决定，询问就是叶子的单点询问。

Problem E 企鹅取暖

- 方法一：将询问离线。
- 对于操作一，建立新点，其儿子分别是 u 和 v 所在树的根
- 对于操作二，建立新点，其儿子为 u 所在树的根。
- 对于操作三，建立新点，其儿子为 u 所在树的根。
- 每个点上携带操作相关信息以及操作的时间戳。
- *dfs* 整棵树，发现每次遇到一个点就是一个子树叶子加，其权值由子树叶子权值和以及父子结点的时间戳决定，询问就是叶子的单点询问。
- *dfs* 序上树状数组即可。

Problem E 企鹅取暖

- 方法二：使用并查集在线维护。

Problem E 企鹅取暖

- 方法二：使用并查集在线维护。
- 额外记录每个结点上次操作的时间戳和一个贡献标记。在每次操作前，先结算这段时间的贡献加入贡献标记，更新时间戳。

Problem E 企鹅取暖

- 方法二：使用并查集在线维护。
- 额外记录每个结点上次操作的时间戳和一个贡献标记。在每次操作前，先结算这段时间的贡献加入贡献标记，更新时间戳。
- 在进行合并时，注意贡献标记的更新，被合并的集合的标记要减去新集合的标记。

Problem F 我想要原石

- 假定以给定点权的点为根，那么考虑一条边权对答案的贡献次数：即这条边向下的子树中点的个数。所以只需要求出子树大小，再经过类似于换根 dp 的 dfs 就可以求出每个点的答案（默认这个点点权为 0 ），做到 $O(1)$ 回答询问。

Problem G 我想玩光遇

- 考虑在 n 秒内累积到队列的人必然会越来越多，所以对于放行策略必然是越靠后放行的人可能越多。

Problem G 我想玩光遇

- 考虑在 n 秒内累积到队列的人必然会越来越多，所以对于放行策略必然是越靠后放行的人可能越多。
- 所以在第 n 秒末一定会放行一次，考虑到放行次数越多越优，故只要间隔一到，就考虑放行，那么整体的放行策略就确定了。

Problem G 我想玩光遇

- 考虑在 n 秒内累积到队列的人必然会越来越多，所以对于放行策略必然是越靠后放行的人可能越多。
- 所以在第 n 秒末一定会放行一次，考虑到放行次数越多越优，故只要间隔一到，就考虑放行，那么整体的放行策略就确定了。
- 显然也可以直接 DP 。

Problem H 排列

- 当 $n = k + 1$ 时，没有操作能够使得满足恰好 k 个位置满足 $i = a_i$。

Problem H 排列

- 当 $n = k + 1$ 时，没有操作能够使得满足恰好 k 个位置满足 $i = a_i$。
- 考虑到不满足条件的位置 i 上的数的目标位置为 a_i，所以可以想到用 DSU 将 i 和 a_i 合并。

Problem H 排列

- 当 $n = k + 1$ 时，没有操作能够使得满足恰好 k 个位置满足 $i = a_i$。
- 考虑到不满足条件的位置 i 上的数的目标位置为 a_i，所以可以想到用 DSU 将 i 和 a_i 合并。
- 对于合并完的每个连通块，若块的大小为 2，则一次操作会增加（减少）2 个满足条件的位置，若块大小大于 2，则一次操作会增加（减少）1 个满足条件的位置。

Problem H 排列

- 当 $n = k + 1$ 时，没有操作能够使得满足恰好 k 个位置满足 $i = a_i$。
- 考虑到不满足条件的位置 i 上的数的目标位置为 a_i，所以可以想到用 DSU 将 i 和 a_i 合并。
- 对于合并完的每个连通块，若块的大小为 2，则一次操作会增加（减少）2 个满足条件的位置，若块大小大于 2，则一次操作会增加（减少）1 个满足条件的位置。
- 为了使操作数尽可能小，优先选择大小为 2 的块。

Problem I 炸弹拆除

- 显然如果单独询问一条边作为路径我们需要最多 1000 次才能访问出答案。

Problem I 炸弹拆除

- 显然如果单独询问一条边作为路径我们需要最多 1000 次才能访问出答案。
- 但是我们每次询问两条相邻的边作为一条路径，我们最多询问 501 次即可得出答案，如果边数为奇，则有一条边会轮空，需要特判。

Problem I 炸弹拆除

- 显然如果单独询问一条边作为路径我们需要最多 1000 次才能访问出答案。
- 但是我们每次询问两条相邻的边作为一条路径，我们最多询问 501 次即可得出答案，如果边数为奇，则有一条边会轮空，需要特判。
- 然后做 dfs 即可，每次询问 u, v 结点，存在两种情况：一是 u, v 有共同的父亲，二是 u 的儿子是 v 的父亲。

Problem J 无限列车

- 我们记所有入度为 0 的点为 $S_1, S_2, S_3, \ldots, S_k$ ，点 u 和点 v 之间的最长路为 $d_{u,v}$ (若不存在一条 u 到 v 的路径，$d_{u,v} = -\infty$)。
  再记 $f_u = \max\limits_{1 \le i \le k} d_{S_i,u}$ 。

Problem J 无限列车

- 我们记所有入度为 0 的点为 $S_1, S_2, S_3, \ldots, S_k$ ，点 u 和点 v 之间的最长路为 $d_{u,v}$（若不存在一条 u 到 v 的路径，$d_{u,v} = -\infty$）。
  再记 $f_u = \max\limits_{1 \le i \le k} d_{S_i,u}$ 。
- 那么在加边前，我们发现点 u 的点权最后会变为 f_u。
  现在考虑如何加边会使所有点的 f 值不变。

Problem J 无限列车

- 我们记所有入度为 0 的点为 $S_1, S_2, S_3, \ldots, S_k$ ，点 u 和点 v 之间的最长路为 $d_{u,v}$（若不存在一条 u 到 v 的路径，$d_{u,v} = -\infty$）。
  再记 $f_u = \max\limits_{1 \le i \le k} d_{S_i,u}$ 。
- 那么在加边前，我们发现点 u 的点权最后会变为 f_u。
  现在考虑如何加边会使所有点的 f 值不变。
- 我们发现一个点只要向 f 值比它大的点连边，就不会影响结果。
  由于允许重边且与加边顺序有关，所以每次加边都是独立的，不会互相影响。

Problem K 这是一场豪赌，朋友！

- 用 (x, y, z) 这样一个三元组来表示一副牌。且 $x + y + z = n$，且 x 表示数量最多的花色对应的牌数，y, z 则分别表示数量第二多和最少的花色对应的牌数，也就是说，我们要求 $x \geq y \geq z$ 。(否则，则对该三元组内部进行排序)。

Problem K 这是一场豪赌，朋友！

- 用 (x, y, z) 这样一个三元组来表示一副牌。且 $x+y+z=n$，且 x 表示数量最多的花色对应的牌数，y, z 则分别表示数量第二多和最少的花色对应的牌数，也就是说，我们要求 $x \geq y \geq z$ 。（否则，则对该三元组内部进行排序）。
- 我们考虑从 (x, y, z) 开始，经过一回合后的可能性：摸两张牌（考虑摸牌顺序，一共有 9 种可能）。对于每种可能性，假设摸完牌后我们得到 $(x+a, y+b, z+c)$，其中，$a+b+c=2$ 。那么对这个新的三元组进行排序，并将数量最少的牌扔掉，我们得到 (x', y', z') 其中 $x'+y'+z'=n$ 且 $x' \geq y' \geq z'$。那么 (x, y, z) 就有 $\frac{1}{9}$ 的可能转移到 (x', y', z') 。

Problem K 这是一场豪赌，朋友!

- 如果我们用 $dp[x,y,z]$ 表示从 (x,y,z) 到摸出暗杠的期望回合数，则可以列出方程:

$$dp[x,y,z]=\frac{1}{9}dp[x_1,y_1,z_1]+\frac{1}{9}dp[x_2,y_2,z_2]+\cdots+\frac{1}{9}dp[x_9,y_9,z_9]+1$$

式子中的 $+1$ 是指这次摸牌经过了一个回合。

Problem K 这是一场豪赌，朋友！

- 如果我们用 $dp[x,y,z]$ 表示从 (x,y,z) 到摸出暗杠的期望回合数，则可以列出方程:

$$dp[x,y,z]=\frac{1}{9}dp[x_1,y_1,z_1]+\frac{1}{9}dp[x_2,y_2,z_2]+\cdots+\frac{1}{9}dp[x_9,y_9,z_9]+1$$

式子中的 $+1$ 是指这次摸牌经过了一个回合。

- 注意在该方程中 (x_i,y_i,z_i) 可能等于 (x,y,z)。则需要移项解方程，具体方法如下：

Problem K 这是一场豪赌，朋友!

- 假设有 k 个 i $(1 \le i \le 9)$ 满足 $(x_i, y_i, z_i) = (x, y, z)$，该方程可表示为：

$$dp[x,y,z] = \frac{k}{9}dp[x,y,z] + \frac{1}{9}dp[x_1,y_1,z_1] + \frac{1}{9}dp[x_2,y_2,z_2]$$

$$+ \cdots + \tfrac{1}{9}dp[x_{9-k}, y_{9-k}, z_{9-k}] + 1$$

Problem K 这是一场豪赌，朋友！

- 假设有 k 个 i $(1 \le i \le 9)$ 满足 $(x_i, y_i, z_i) = (x, y, z)$，该方程可表示为：

$$dp[x,y,z] = \frac{k}{9}dp[x,y,z] + \frac{1}{9}dp[x_1,y_1,z_1] + \frac{1}{9}dp[x_2,y_2,z_2]$$

$$+ \cdots + \tfrac{1}{9}dp[x_{9-k}, y_{9-k}, z_{9-k}] + 1$$

- 解方程可得：

$$dp[x,y,z] = \frac{1}{9-k}dp[x_1,y_1,z_1] + \frac{1}{9-k}dp[x_2,y_2,z_2]$$

$$+\cdots + \tfrac{1}{9-k}dp[x_{9-k}, y_{9-k}, z_{9-k}] + \tfrac{9}{9-k}$$

Problem K 这是一场豪赌，朋友！

- 假设有 k 个 i $(1 \le i \le 9)$ 满足 $(x_i, y_i, z_i) = (x, y, z)$，该方程可表示为：

$$dp[x,y,z] = \frac{k}{9}dp[x,y,z] + \frac{1}{9}dp[x_1,y_1,z_1] + \frac{1}{9}dp[x_2,y_2,z_2]$$

$$+ \cdots + \tfrac{1}{9}dp[x_{9-k}, y_{9-k}, z_{9-k}] + 1$$

- 解方程可得：

$$dp[x,y,z] = \frac{1}{9-k}dp[x_1,y_1,z_1] + \frac{1}{9-k}dp[x_2,y_2,z_2]$$

$$+ \cdots + \tfrac{1}{9-k}dp[x_{9-k}, y_{9-k}, z_{9-k}] + \tfrac{9}{9-k}$$

- 后续用记忆化搜索即可。

Problem L 饿了吧！

- 二进制枚举每个月饼买或不买，求出最优解即可。

Thank You!