浙江工业大学程序设计迎新赛决赛 - 简要题解

浙江工业大学程序设计竞赛命题组

ZJUT

2022 年 3 月 6 日

Problem A. 水质检测

- 若给定序列的最大值小于等于 k，输出 Good。
- 若给定序列的平均值小于等于 k，输出 OK。
- 否则输出 Bad。

Problem H. 基础离散数学练习题

- 对于给定数列 x_i，记 cnt_a 为 a 在给定数列中出现的次数，题目实际上就是让我们找最大的 i 使得 $cnt_i = i$，如果不存在则为 -1。

Problem F. DiDiDi 与孪生素数

- 不妨假设孪生素数为 $(x-1, x+1)$，其中 $x = 6n+m\ (0 \le m < 6)$。
- 要求 $x-1 = 6n+m-1$ 和 $x+1 = 6n+m+1$ 为素数，在 $n > 0$ 的前提下，不难得出 m 只能取 0，即和一定为 12 的倍数。
- 特别地，当 $n = 0$ 时，存在孪生素数 $(3, 5)$，其和为 8，需要特判。

Problem B. MS 与美食街 2

- 首先，注意到每一份花费都形如 a^{c_i}，为了简化问题，我们不妨在 a 进制下考虑。
- 要求出最少要携带多少钱，我们需要考查这 n 个数的和，在 a 进制下可以通过模拟进位过程来实现。
- 由于 a 很大，我们并不能从低到高枚举每一位模拟进位的过程，但是 n 很小，可以知道进位的次数不会很多，于是我们可以使用 std::map 来模拟这一过程。

Problem B. MS 与美食街 2

- 知道了在 a 进制下这 n 份花费的和以后，我们需要最小的大于总花费的形如 a^k 的数，这个问题就非常简单了。
- 如果总和在 a 进制下形如 $(\overbrace{10000\cdots}^{k\text{ 位}})_a$，那么所求最小花费为 a^k，否则答案为 a^{k+1}。

Problem J. Lecxcy 与位运算

- 对于按位异或运算，位与位之间独立互不影响，于是可以单独考虑每一位对答案的贡献，从而对问题进行简化。
- 单独考虑第 i 个二进制位时，每个数只能取值于 $\{0, 1\}$，此时要计算 $\sum_{1\le i<j\le n} a_i \oplus a_j$。
- 结合异或运算的性质，我们只关心 0 和 1 的个数，不妨分别记为 $\mathrm{cnt}_{i,0}$ 和 $\mathrm{cnt}_{i,1}$，那么它对于最终答案的贡献为 $2^i \cdot \mathrm{cnt}_{i,0} \cdot \mathrm{cnt}_{i,1}$，如此对每个二进制位进行统计和累加即可。

Problem I. 翻牌

- 我们首先证明这个游戏一定会结束，把排成一列的 n 张牌用 n 位二进制数表示，我们每一次翻牌都必须使得所选区间最左侧的 1 变为 0 ，这表明这个二进制数必定一直在减小，因此有限个回合后游戏终止。
- 我们再说明本题的结论：容易发现我们能够选择的左端点必定 $\leq n-k+1$ ，我们依次选择 $n-k+1, n-2k+1, \cdots, n-\lfloor \frac{n}{k} \rfloor k+1$ 这几个位置，然后记函数 $f(n,k)$ 为这些位置的状态之和（正面贡献为 1 ，背面为 0）。

Problem I. 翻牌

- 该函数的初值显然是 $\lfloor \frac{n}{k} \rfloor$ ，同时根据抽屉原理不难发现这个函数在相邻两次翻牌操作的函数值正好变化 1 ，即 $|f_{i+1} - f_i| = 1$ ，因此先后手不论怎么翻牌，该函数的函数值奇偶性是固定的，如果某个人固定是奇数，那就说明他一定能够翻牌，即必胜。
- 综上所述，$\lfloor \frac{n}{k} \rfloor$ 为奇数时先手必胜，偶数时后手必胜。

Problem E. 招募军队

- 要求最大化 c 与 $\sum_{i=1}^{n} a_i$ 差值的绝对值，实际上就是要我们求出 c 的最大值和最小值。
- 于是题目转化为：给定 n 个点，m 条边的无向图，将其分成若干连通块，每个连通块满足无环并且有且仅有一个点在自己的城市招募军队。
- 考虑引入一个超级源点 S，S 与每个点 i 连接权值为 a_i 的边，即转化为新图上的最小/最大生成树问题。

Problem K. KoiKoi∼

- 按照题意模拟，对给出的役种进行统计即可。

Problem D. MS 与矩阵取数

- 首先先给九宫格编个号。

1	2	3		5	1	4
4	5	6	→	1	9	1
7	8	9		9	8	1

- 以样例为例，先以 2、4、6、8 为中心，使他们四个位置上的数相等（标红的是选择的位置）。

5	1	4		5	1	4
1	9	1	→	1	2	1
9	8	1		2	1	4

Problem D. MS 与矩阵取数

- 然后考虑以 1、6、8 为中心，同时选择这三个位置，重复多次使得 1、9 两个位置上的数相等。显然同时选择 1、6、8 后，2、4、6、8 四个位置上的数仍相等，但位置 1 会上的数只增加 1，而位置 9 上的数会增加 2，所以只要重复几次就能使他们相等。

5	1	4		6	2	5
1	2	1	→	2	4	2
2	1	4		3	2	6

Problem D. MS 与矩阵取数

- 同理再同时选择 3、4、8 为中心，重复多次使得 3、7 两个位置上的数相等。

6	2	5		8	4	7
2	4	2	→	4	8	4
3	2	6		7	4	8

- 最后同时选择 3、7 为中心，重复选择使得 1、3、7、9 四个位置上的数相等。

8	4	7		8	5	8
4	8	4	→	5	8	5
7	4	8		8	5	8

Problem D. MS 与矩阵取数

- 然后选择 1、3、5、7、9 为中心，使得 2、4、6、8 上的数和位置 5 的数的差是 3 的倍数（如果已经是 3 的倍数就不用操作）。
- 然后选择 2、4、6、8 为中心，让 2、4、6、8 上的数和位置 5 的数相等，因为这一步可以让 2、4、6、8 上的数 +1，而中间的数 +4，所以上一步要让他们的差为 3 的倍数。

8	5	8		6	4	6
5	8	5	→	4	4	4
8	5	8		6	4	6

- 然后只要一直选中间的数即可。

Problem C. MS 与矩阵填数

- 注意到每个数的贡献独立，考虑每一个数作为行最小数的方案数。
- 假设选定 i 为行最小值，首先我们要求同一行的数都要比他大，确定一行中所有数字的方案数为 $\binom{n^2-i}{n-1}$。
- 这一行内部的数可以随意排列，方案数为 $n!$，且这一行可以是 n 行中的任意一行，并且其他行的数有 $(n^2-n)!$ 的填法。
- 于是答案为 $\sum_{i=1}^{n} i \cdot n \cdot n! \cdot (n^2-n)! \cdot \binom{n^2-i}{n-1}$。

Problem M. 智慧交通

- 由于限制了所有询问中不用上课的同学数量之和为 $S = 10^5$，不难发现不上课人数大于 $\sqrt{S}$ 的询问数不多于 $\sqrt{S}$。
- 可以考虑以 $\sqrt{S}$ 为阈值对询问规模根号分治，若不上课的人数大于阈值，我们对所有上课的人倒着跑最长路，最多需要这样做根号次，时间复杂度 $\mathcal{O}(n\sqrt{n})$。

Problem M. 智慧交通

- 若不上课的人数不大于阈值，可以对每个点作为上课地点的情况预处理它前面的距离它前根号远的点的距离，在这根号个结果中去除没上课的同学的情况，就可以求出答案。具体预处理可以在拓扑排序的过程中维护合并以后的距离集合，预处理和询问时间复杂度均为 $\mathcal{O}(n\sqrt{n})$。

Problem L. 骰子 2

- 在这个问题中，我们需要关注 n 个六面骰子和 $\frac{n}{2}$ 个十二面骰子掷出点数的和的分布情况，以下讨论前者的处理方法，后者同理。
- 定义 $f_{n,i}$ 表示 n 个六面骰子掷出和为 i 的方案数，考虑一个骰子的情形，有普通生成函数

$$F_1(x) = \sum_{i=1}^{6} f_{1,i} \cdot x^i = \sum_{i=1}^{6} x^i = \frac{x - x^7}{1 - x}.$$

- 而 n 个骰子的普通生成函数就是对 1 个骰子情形做 n 次卷积

$$F_n(x) = \left(\frac{x - x^7}{1 - x}\right)^n.$$

Problem L. 骰子 2

- 对分子和分母使用广义二项式定理展开后，使用 NTT 计算卷积求出 $F_n(x)$ 对应项系数，就得到了两种骰子掷出结果的分布规律，于是答案就不难计算了。
- 直接使用卷积快速幂求 $F_n(x)$ 也可以通过本题。

Problem G. 圆圆 circle

- 假设所有选择初始时均为顺时针，现在要翻转若干个操作区间，要保证最优性，可以肯定这若干个区间未翻转前一定两两相交。如果存在两个区间不交，显然不是全部反转他们结果最优。
- 那么假定全部顺时针操作每个点的圆环数是 a_i，反转一些区间之后每个点的圆环数是 b_i，并且这些区间交的区间为 $[l, r]$，我们在这个区间中取出一个最大的 b_i 记为 b_{pos}。

Problem G. 圆圆 circle

- 由于所有区间交内的点都原来被所有未翻转区间覆盖，现在又都不被所有未翻转区间覆盖，他们的差值变化是相同的。
  1. $\forall i \in [l, r], a_i - b_i = a_{pos} - b_{pos}$ 。
- 所有区间交之外的点至少不被一条未翻转区间覆盖，并且翻转后会被该区间覆盖，所以差值至少比区间交之内的点少 2。
  2. $\forall i \notin [l, r], a_i - b_i \leq a_{pos} - b_{pos} - 2$。
- 因为交区间内的差值全部一致，并且 b_{pos} 是区间内最大的，所以 a_{pos} 是区间内最大的。
  3. $a_{pos} = \max\{a_i\}, i \in [l, r]$。

Problem G. 圆圆 circle

- 其次有一个不是很显然的性质：
  4. 存在一个最优方案满足，$b_{pos} \geq \max\{b_i\} - 1, i \in [1, n]$。
- 考虑反证法证明这个性质，如果 $b_{pos} < \max\{b_i\} - 1, i \in [1, n]$，那么我们可以选择左端点最大的一个反转区间，和一个右端点最小的反转区间，让他们不反转，这样子 $[l, r]$ 之间覆盖全部加二，而左端点最大和右端点最小之外的所有区间覆盖全部减二，其他不变。也就是最大值至少不变，但是 b_{pos} 变大 2，如此进行这些操作，最后 b_{pos} 会变成比 $\max\{b_i\}$ 小的最大的原数加二得到的值，所以存在一个最优方案是满足性质 4 的。

Problem G. 圆圆 circle

- 结合性质 2、3、4 可以推出性质 5。
  5. 存在一个最优方案满足，$a_{pos} \geq \max\{a_i\}, i \in [1, n]$。
- 因为 $\forall i \notin [l, r], a_i - b_i \leq a_{pos} - b_{pos} - 2$ 并且 $b_{pos} \geq max\{b_i\} - 1, i \in [1, n]$。组合一下可以知道 $b_{pos} + (a_{pos} - b_{pos} - 2) \geq b_i - 1 + (a_i - b_i)$，消元得 $a_{pos} \geq a_i + 1$。

Problem G. 圆圆 circle

- 也就是说，**一开始在全部顺时针排布中覆盖圆环数最多的位置** a_k**，肯定存在一种方案使得它成为最优方案中的** b_{pos}。
- 考虑二分这个值变成了多少，假设变成了 res，那么根据性质 4，可以知道最大值要么是 res，要么是 $res+1$，记这个最大值为 x。并且此时总共反转的区间个数是可知的，我们假定这一部分为 cnt。
- 对于每一个点 i，有 x_i 个经过他的区间反转了，那么显然还有 $cnt - x_i$ 个反转区间没有经过它，于是最后满足 $a_i - x_i + cnt - x_i < res$，这样一来就可以二分了，对于一个点，可以考虑从最远的区间开始（用堆维护），把这个点反转到覆盖的区间小于限制的最小情况，然后检查这样子会不会出现有区间矛盾即可。