Solutions For NOIP 模拟 Day 4

Claris

2021 年 8 月 2 日

循环卷积

给定两个数组 $a[0..n-1], b[0..n-1]$，求出它们的循环 Max+ 卷积。

测试点 1,2,3

- $n = 1$。
- $c_0 = a_0 + b_0$。
- A + B Problem！

测试点 4,5,6

- $n \leq 100$。
- 按照定义计算循环卷积即可。
- 时间复杂度 $O(n^2)$。

测试点 7,8,9,10

- $0 \leq a_i, b_i \leq 5000$，$\sum a_i \leq 5000$，$\sum b_i \leq 5000$。
- 令 A, B 分别为两个数组的最大值，则 $c_i \geq \max(A, B)$。
- 另一方面，如果 $c_i > \max(A, B)$，则对应的 a_i 和 b_j 都非零。
- 根据题意，两个数组最多只有 5000 个数非零，暴力计算即可。
- 时间复杂度 $O(n + 5000^2)$。

异或的平方和

给定 n 个正整数 $a_1, a_2, \ldots, a_n$ ，请求出：

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(a_i \oplus a_j)^2$$

测试点 1,2,3

- $n \le 1000$。
- 暴力枚举 i 和 j，计算对答案的贡献。
- 时间复杂度 $O(n^2)$。

测试点 4,5,6

- $a_i \le 1000$。
- 暴力枚举 a_i 和 a_j 的值，将贡献乘以方案数后累加到答案中。
- 时间复杂度 $O(n + a^2)$。

测试点 7,8,9,10

- 将 a_i 和 a_j 按二进制每一位展开：
- $$a_i = \sum_{x=0}^{29} 2^x a_{i,x}$$
- $$a_j = \sum_{y=0}^{29} 2^y a_{j,y}$$
- $$a_i \oplus a_j = \sum_{k=0}^{29} 2^k [a_{i,k} \neq a_{j,k}]$$
- 将展开式代入原式。

测试点 7,8,9,10

- $$\begin{aligned} & \sum_{i=1}^{n}\sum_{j=1}^{n}(a_i \oplus a_j)^2 \\ = \quad & \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\sum_{x=0}^{29} 2^x[a_{i,x} \neq a_{j,x}]\right)\left(\sum_{y=0}^{29} 2^y[a_{i,y} \neq a_{j,y}]\right) \\ = \quad & \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{x=0}^{29}\sum_{y=0}^{29} 2^{x+y}[a_{i,x} \neq a_{j,x}][a_{i,y} \neq a_{j,y}] \\ = \quad & \sum_{x=0}^{29}\sum_{y=0}^{29} 2^{x+y}\sum_{i=1}^{n}\sum_{j=1}^{n}[a_{i,x} \neq a_{j,x}][a_{i,y} \neq a_{j,y}] \end{aligned}$$

测试点 7,8,9,10

- $$\sum_{x=0}^{29}\sum_{y=0}^{29}2^{x+y}\sum_{i=1}^{n}\sum_{j=1}^{n}[a_{i,x} \neq a_{j,x}][a_{i,y} \neq a_{j,y}]$$
- 枚举 $x, y, a_{i,x}, a_{j,x}, a_{i,y}, a_{j,y}$ 的取值，乘以对应的方案数（即有多少个 i 满足第 x 位为 $a_{i,x}$ 且第 y 位为 $a_{i,y}$）即可。
- 预处理出 $f_{i,j,x,y}$ 表示有多少个数满足第 i 位为 x 且第 j 位为 y，则方案数可以查表 $O(1)$ 得到。
- 时间复杂度 $O(n\log^2 a)$。

景点距离

给定一棵 n 个点的完全二叉树，每条边长度都是 1。

m 次操作，每次要么删掉一条边，要么询问有多少点对仍然连通且最短路不超过 k。

测试点 1,2,3

- $n \le 50, m \le 50$。
- 对于每个操作，暴力统计有多少点对仍然连通且最短路不超过 k。
- 时间复杂度 $O(mn^2)$。

测试点 4,5,6

- 没有删边操作。
- 树形 DP，设 $f_{i,j}$ 表示 i 子树里有多少个点到 i 的距离恰好为 j。
- 再计算出 $g_{i,j}$ 表示 i 子树外有多少个点到 i 的距离恰好为 j。
- 自底向上推出 f，再自顶向下推出 g。
- 时间复杂度 $O(nk)$。

测试点 7,8,9,10

- 记录每条边被删除的时间。
- 设 $cnt_{i,j}$ 表示有多少边数恰好为 j 的路径的删除时间的最小值恰好为 i。
- 那么对于第 i 个询问，$ans = \sum cnt_{\geq i, \leq k}$，对 cnt 求出前缀和即可。

测试点 7,8,9,10

- 考虑如何求出 $cnt_{i,j}$。
- 枚举路径的最近公共祖先 x，求出 x 的所有子节点 u 到 x 路径上的删除时间最小值 a_u 以及经过的边数 b_u。
- 将所有点 u 按照 a_u 从大到小排序。
- 依次考虑每个 u 的 a_u 作为路径最小值的情况。
- 设 s_j 表示在 u 之前有多少点 v 满足 $b_v = j$，枚举所有 j，对 $cnt_{a_u,j+b_u}$ 的贡献为 s_j。
- 完全二叉树的情况下总时间复杂度为 $O(n\log^2 n + m\log n)$。

数字重组

给定 n 个正整数和一个正整数 k，你需要将这 n 个整数重组成 k 个集合，每个集合恰好 $\frac{n}{k}$ 个数，使得每个集合内都没有重复的数字。

请找到一个数字重组方案，使得该方案里这 k 个集合的极差之和最小。

测试点 1,2

- $n \le 6$。
- 暴力搜索每个数划分到哪个集合即可。
- 时间复杂度 $O(k^n)$。

测试点 3,4,5,6

- $n \le 16$。
- $O(2^n)$ 枚举所有可能的集合 S，判断 S 是否存在重复数字，并预处理出 v_S 表示 S 的极差。
- 子集 DP，设 f_S 表示将 S 集合的数进行合法分组的最小极差和。
- $f_S = \min(f_{S-T} + v_T)$，其中 T 是 S 的子集且 T 不存在重复数字。
- 时间复杂度 $O(3^n)$。

测试点 7,8,9,10

- 从小到大依次考虑每个数字，一开始 k 个集合都是空的。
- 考虑到数字 x 的时候，枚举将其放入哪个集合，那么那个集合的元素数量应该小于 $\frac{n}{k}$。
- 如果对应集合为空，那么 x 对总极差的贡献为 $-x$。
- 如果对应集合元素数量为 $\frac{n}{k}-1$，那么 x 对总极差的贡献为 x。

测试点 7,8,9,10

- 设 $f_{i,S}$ 表示考虑了前 i 小的数字，每个集合的元素数量情况为 S 的最小总极差，其中 S 是个长度为 k 的 `std::vector`，依次表示每个集合的元素个数。
- 注意到交换两个集合的元素个数不影响答案，所以可以对 S 进行排序。
- 每个状态 $f_{i,S}$ 都能看作从 $(0,0)$ 出发往右往上走到达 $(k,\frac{n}{k})$ 的折线。
- 状态数为 $C(k+\frac{n}{k},k) \le C(2\sqrt{n},\sqrt{n})$。
- 如何保证每个集合都没有重复元素？

测试点 7,8,9,10

- 修改状态的定义。
- 设 $f_{i,j,S}$ 表示考虑了前 i 小的数字，每个集合的元素数量情况为 S 的最小总极差，其中第 i 小的数字所在的集合在放入 i 之前的元素数量为 j。
- 从 $i-1$ 转移到 i 时，若 $a_i = a_{i-1}$，那么只要保证 i 放入的集合元素数量不超过 j 即可满足每个集合没有重复元素。
- 这是因为之前有 a_i 的集合至少有 $j+1$ 个元素。
- 一共有 $O(\frac{n}{k} \times C(2\sqrt{n}, \sqrt{n}))$ 个状态，每个状态朴素转移的复杂度为 $O(k^2)$。
- 总时间复杂度 $O(\frac{n}{k} \times C(2\sqrt{n}, \sqrt{n}) \times k^2) = O(nkC(2\sqrt{n}, \sqrt{n}))$。

Thank you!