

CSP-S 模拟赛题解

AutumnKite

2023 年 10 月 6 日

1 切纸带

1.1 算法一

枚举左端点，然后在枚举右端点时，每次遇到和等于 m 就切成一段。
时间复杂度 $O(n^2)$ 。期望得分 60 分。

1.2 算法二

令 f_i 表示以 i 为右端点最多能切多少段。暴力做法是往前找到 j 使得 $j+1$ 到 i 的和为 m ，然后令 $f_i = f_j + 1$ 。

可以二分找到 j ，或在枚举 i 的同时维护 j （即双指针）即可。时间复杂度 $O(n \log n)$ 或 $O(n)$ 。期望得分 100 分。

2 排列计数

2.1 算法一

枚举所有排列判断即可。时间复杂度 $O(n! \cdot \text{poly}(n))$ ，期望得分 10 ~ 20 分。

2.2 算法二

状压 DP，令 $f_{S,i,j}$ 表示已经用了 S 中的数，以 i 结尾，满足题中条件的位置有 j 个的方案数。时间复杂度 $O(2^n \cdot \text{poly}(n))$ ，期望得分 30 ~ 40 分。

2.3 算法三

考虑钦定若干个位置满足条件，则这些位置构成了若干个连续段，每个连续段一定由一段连续递增的数或连续递减的数构成。可以使用 DP 求出钦定了 i 个位置的答案。

每次询问使用容斥求出恰好 k 个位置满足条件的方案数即可。

时间复杂度 $O(n^2 + Tn)$ ，期望得分 80 分。

2.4 算法四

考虑从 1 到 n 依次插入。假设当前有 1 到 i 的排列，要插入 $i+1$ ，有以下几种情况：

1. i 与 $i-1$ 相邻且插在 i 和 $i-1$ 之间。
2. 插在 i 的两侧，但不是在 i 和 $i-1$ 之间。
3. 插在一对值相邻的数之间，且不是 i 的两侧。

4. 插在一对值不相邻的数之间，且不是 i 的两侧。

令 $f_{i,j,0/1}$ 表示 1 到 i 的排列，有 j 对数相邻， i 与/不与 $i-1$ 相邻的方案数。转移考虑上述四种情况即可。

预处理 f 以后即可 $O(1)$ 查询。时间复杂度 $O(n^2 + T)$ 。期望得分 100 分。

3 单向道路

3.1 算法一

$n = 2$ 时输出 -1 。期望得分 5 分。

3.2 算法二

暴力搜索，期望得分 15 ~ 30 分。

3.3 算法三

考虑特殊性质。

首先可以注意到，将题目改成每次加入一条反向边（即保留原图中的边），答案并不会改变。

接下来有两种做法：

- 考虑将原图的一条边看成一条线段，相当于选择若干线段覆盖 1 到 n 。暴力 DP 时间复杂度 $O(n^3)$ ，可以在枚举转移点的同时维护后缀 \min ，优化至 $O(n^2)$ 。
- 考虑建一张新图，对于所有 $i < j$ ，建立 $(i, j, 0)$ 和 $(j, i, a_{i,j})$ 两条边，则答案为 n 到 1 的最短路。使用 Dijkstra 解决，时间复杂度 $O(n^2 \log n)$ 。

期望得分 15 分。

3.4 算法四

有了算法三中的结论，我们可以直接将原图的每个强连通分量缩成一个点，转化为特殊性质的情况。

竞赛图的缩点不需要使用 Tarjan，可以将所有点按入度从小到大排序，以所有前缀和等于 $\frac{i(i-1)}{2}$ 的位置分隔即可。

然后采用算法三中任意一种做法即可。期望得分 100 分。

事实上，对于算法三中第二种做法，可以略去缩点的步骤，只需要找到首尾两个强连通分量中任意一个点作为起点和终点。选取入度最小和最大的点即可。

4 二叉搜索树

4.1 算法一

模拟题目中的操作。时间复杂度 $O(q^2n)$ ，期望得分 15 分。

4.2 算法二

考虑将插入每个 BST 的数存下来，每次询问的时候暴力建树查询。时间复杂度降为 $O(q(n+q))$ ，期望得分 30 分。

4.3 算法三

考虑 $n = 1$ 的情况。我们将操作离线，并把权值离散化。

令 t_i 表示权值为 i 的节点插入的时间，则最终的 BST 是序列 t 的笛卡尔树。

对于询问的权值 x ， x 在笛卡尔树上的所有祖先即为以 x 结尾的前缀的所有后缀最小值的位置以及以 x 开头的后缀的所有前缀最小值的位置。

考虑对前后缀分别求和。以前缀为例，我们使用单调栈维护所有后缀最小值的位置，然后在单调栈上二分出该询问对应的时间，单调栈中该位置的前缀即为所有这个时间存在的祖先。

时间复杂度 $O(q \log q)$ ，期望得分 45 分。稍作修改可以通过测试点 10 ~ 11，期望得分 55 分。

4.4 算法四

仍然考虑离线，并将对一个区间的插入操作差分为在左端点的插入和在右端点的删除。

然后我们从左往右扫描每棵 BST，并使用线段树维护这个 BST 对应的 t 序列即可。

对于数据随机的情况，我们可以在线段树上维护单调栈，期望时间复杂度 $O(q \log^2 q)$ ，结合算法三，期望得分 75 分。

4.5 算法五

对于一般情况，我们使用类似“楼房重建”的做法，在线段树 pushup 时递归右儿子得到更新后的信息。

时间复杂度 $O(q \log^2 q)$ ，期望得分 100 分。