Solution Day1

2023 年 9 月 28 日

1 stamps

从 N 个区间中选择 K 个区间，问最多覆盖多长的长度。

设 $dp_{i,j}$ 表示到 i 这个位置，选择了 j 个区间时，能覆盖的最长长度

- 如果选择覆盖 i，一定会选右端点最远的区间，转移到 $dpk, j+1$
- 如果不覆盖 i，则转移到 $dp_{i+1,j}$

2 color

先考虑是一棵树的情况，每个节点对色块数的贡献，答案为每个节点边的颜色数之和-(n-1)，对于每次修改操作，用 map 统计节点的增量即可。基环树是一个环上每一个节点连接了一棵树，和树的情况类似。答案为每个节点边的颜色数之和 - n。同时要注意，如果整个环上只有一种颜色，答案还要 + 1。

3 tree

首先，当蓝色路径变成红色边时，不会导致比以前更多的接触情况。初始状态是一棵有向根树。如果在最终状态中存在一条边进入非叶子顶点 v，那么必须也存在一条从 v 出发的边，因此它可以合并为一条边。因此，存在一种最优的最终状态，其中每条边的端点都是初始树的叶子顶点。

考虑到这一点，问题是将有向根树的边分为路径 (u, v)，其中 v 应该是初始树的叶子，目标是最小化 $\sum C_u C_v$。

如果有一条红色边 (u, v) 是最终状态，我们称 v 与 u 相连。考虑一个不是根的顶点 u。我们称以 u 为根的子树为 T_u。那么 T_u 中一定存在一个叶子节点与 u 的祖先相连。

基于此，我们可以考虑以下动态规划解决方案：

设 $D[u][l]$: 当 l 与 u 的祖先相连时可以达到的最小成本。

$D[u][l] = \min(D[x][l] + P(D[c][l_c] + C_u C_{l_c}))$，其中 x 是 l 的祖先的 u 的子节点，c 是 x 的所有其他子节点，l_c 是使 $D[c][l_c] + C_u C_{lc}$ 最小的叶子节点。

对于 T_u 中的每个叶子 l，考虑一条 $y = C_l \times x + D[u][l]$ 的直线。我们可以看到在上述动态规划公式中只使用了这些直线的下凹壳。如果对于 u 的每个子节点 c，我们都有这些直线的下凹壳，那么我们可以计算 T_u 的下凹壳。

我们可以看到，组成 T_u 下凹壳的线是 u 的子节点 c 的下凹壳的并集，沿着 y 轴方向移动。

由于合并大小为 A 和 B 的两个下凹壳可以在 $O(\min(A, B) \log \min(A, B))$ 的时间内完成，借助类似 BST 的数据结构，整个过程可以在 $O(N \log^2 N)$ 的时间内完成。

由于李超树做线段树合并最后的总复杂度是 $O(N \log N)$ 的。又发现在 DP 的式子当中只会修改 $y = kx + b$ 中的 b ，也就是说相当于对于一颗子树中的下凸壳做了平移。在线段树合并的过程中对子树凸壳打一个整体平移的标记即可实现 $O(N \log N)$。

4 string

在模拟 Manacher 的过程中，我们能确定出 $O(n)$ 个字符相等与不相等的信息，这些信息可以完美包揽了 manacher 的信息。

为了更好地维护这些信息，我们将相同字符的点缩点，不相同的点连边，形成了一张无向图。

Sa 的信息是诱导排序能给我们的，利用诱导排序，即比较 sa_{i-1}, sa_i 时利用 $rk_{sa_{i-1}+1}, rk_{sa_i+1}$ 的信息，当两者符号相反时则直接确定了严格小于的信息，否则确定了大于等于的信息。

而 Manacher 的不等信息也可以利用 Sa 里的 rk 来确定偏序关系。于是我们将无向图转化成有向图，实际上这是一张偏序关系形成的图，它恰好包含了 Manacher 和 Sa 的所有信息。

通过 Tarjan 缩点，然后在新图上用拓扑排序求每个点离起点的最长路来确定该点的下界，这样就可以贪心地求出了答案。

实际上 Tarjan 缩点时是可以直接判断，不过怕麻烦的可以在构造出答案后再用 Manacher 和 Sa 求出答案。时间复杂度 $\mathcal{O}(n)$.