联考废案题解

Xun_Xiaoyao

2023 年 10 月 10 日

1 树状数组 (fenwick)

一道很有意思的签到题!

1.1 测试点 $1 \sim 8$

使用题目给出的函数进行 1 操作，二操作暴力查区间和，时间复杂度 $O(Qn)$。

1.2 测试点 $9 \sim 14$

使用线段树或者树状数组维护区间和，由于一次 1 操作有 $O(\log n)$ 次修改，时间复杂度 $O(Q \log^2 n)$。

1.3 测试点 $15 \sim 20$

考虑利用要树状数组的结构，考虑使用树状数组维护前缀和，每一次 1 操作只会修改 $O(\log n)$ 个数，单次修改可以做到 $O(\log n)$，时间复杂度 $O(Q \log n)$。

2 子序列 (seq)

sto moonstaring orz

2.1 测试点 $1 \sim 4$

枚举所有的子序列，时间复杂度 $O(2^n \operatorname{poly}(n))$

2.2 测试点 $5 \sim 8$

考虑枚举 A 有多少个子序列恰好在第 i 位比 B 小，前若干位匹配可以贪心，然后枚举 A 的第 i 位是什么数，使用 DP 求出在后面有多少本质不同的子序列，做到 $O(\operatorname{poly}(n))$

2.3 测试点 $9 \sim 20$

考虑优化上述做法，记 $f_{i,j}$ 表示在 A 的 $[i,n]$ 这段后缀中，以 j 为第一个数的子序列的数量，不难得到转移 $f_{i,j} = \begin{cases} \sum\limits_{k=1}^{V} f_{i+1,k} + 1 & ,j = a_i \\ f_{i+1,j} & ,otherwise \end{cases}$。

考虑在 A 中贪心匹配 B（每次取最前的一位），假设匹配的位置分别是 $c_1, c_2 \dots c_m$，则最终答案按为 $\sum\limits_{i=1}^{m} \sum\limits_{j=1}^{b_i-1} f_{c_i,j} + 1$，发现这个式子可以用树状数组优化，时间复杂度 $O(n \log n)$。

3 新建文件夹 (build)

sto Fyfive,Conviction orz

把 n, m, q 视为同阶。

3.1 测试点 $1 \sim 4$

暴力维护一个 $n \times m$ 的矩阵即可，暴力求值，时间复杂度 $O(qnm)$。

3.2 测试点 $5 \sim 10$

维护二维前缀和，时间复杂度 $O(nm + q)$。

3.3 正解 1

二维矩阵和板子题。

使用线段树维护区间历史版本和。

具体的你可以实时维护一个一次函数 $kt + b$ 表示当前值为 k，当前时刻 t_0 的历史版本和为 $kt_0 + b$。

我们发现，如果这个值没有被修改，历史版本和可以直接顺延为 $k(t_0 + 1) + b$，也就是这个一次函数。

否则，对这些一次函数维护懒标记修改即可。

可以做到 $O((m + q) \log n)$。

3.4 正解 2

线段树的节点维护向量 $\begin{bmatrix} ans \\ nw \\ len \end{bmatrix}$，表示历史版本区间和 ans，当前值区间和 nw 以及区间长度 len。

则一次区间修改就是给这个区间右乘一个矩阵 $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & x & 1 \end{bmatrix}$。

记录一次历史版本是右乘一个矩阵 $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$。

时间复杂度 $O(3^3(m + q) \log n)$，常数略大。

发现只有下三角的 6 个数可能不为 0，只要维护这 6 个数即可，常数会小一点。

4 树题 (tree)

sto Ustinian505 orz

4.1 测试点 $1 \sim 4$

对于每个 u 暴力枚举 $v \in sub_u$，然后 $O(n)$ 计算答案。

时间复杂度 $O(n^3)$，期望得分 20 分。

4.2 测试点 $5 \sim 6$ 和 $11 \sim 14$

考虑枚举 v，在 T_2 上暴力跳祖先维护第二类贡献，同时使用 $O(1)/O(\log n)$ 求 lca 求两点间路径长度，更新祖先的答案。

时间复杂度 $O(\sum dep)/O(\sum dep \log n)$，期望得分 50 分。

4.3 测试点 $7 \sim 10$

第一类贡献 $dis(u,v)$ 是序列上的，直接上分治。

当前解决 $[l,r]$，要计算 $[l,mid],[mid+1,r]$ 相互的贡献。设 w_i 为 i 到 mid 的距离。

需要最小化 $w_i + w_j + \sum A_{P(j \to i)_k} \times (k-1)$。

后半部分贡献不好看，转换一下：

设

$$C_i = \sum_{j \in sub_i} A_j, D_i = \sum_{j \in sub_i} dep_j \times A_j$$

那么

$$\sum A_{P(j \to i)_k} \times (k-1) = D_{g_i} - dep_j \times C_{g_i} + dep_v \times C_j - D_j$$

原式变为 $(w_i + D_{g_i}) - dep_j \times C_{g_i} + (dep_j \times C_j - D_j + w_j)$。

对于每个 i，$w_i + D_{g_i}$ 是确定的，需要最小化后面的乘积式，同时需要满足 i,j 分属 mid 两侧且 $j \in sub_i$。

- $j \in sub_i$，将 $[l,r]$ 内的点拉出来建**虚树**。
- 最小化乘积式，我会**李超树合并**。
- i,j 分属 mid 两侧，每个点维护分别维护 $\le mid$ 和 $> mid$ 的李超树。

时间复杂度 $O(n \log^2 n)$，拼上暴力后期望得分 70 分。

4.4 测试点 $15 \sim 20$

将序列分治换成**边分治**即可，其余方法不变。

为什么不用**点分治**？因为每个分治中心的不同子树可能很多，那么每个点需要维护子树个数个李超树，时间复杂度就错了。（如果你会用点分治解决，那就太厉害了!）

时间复杂度 $O(n \log^2 n)$，期望得分 100 分。