

1 神奇的二叉查找树 (bst.cpp/c/pas)

题解

由于是按顺序插入二叉查找树中的，所以原来在某个节点的左孩子中的一系列点和在该节点右孩子中的一系列点的顺序是无关紧要的，可以用组合数处理出来。

我们可以模拟二叉查找树的建树，每次找到根，并将剩下的数划分成左孩子和右孩子，这两类数之间的顺序可以更改。然后递归进行操作即可。时间复杂度 $O(mn^2)$ 。

2 神奇的字母 (alpha.cpp/c/pas)

题解

将 a 赋权值 1，将 b 赋权值 2；定义字符串的权值 $w(s)$ 为其中所有字母权值的和。然后我们接下来在权值对 3 取模的意义下操作。

一个字符串 s 可以经过若干次操作变为字符 c ，当且仅当：

1. $w(s) = w(c)$;
2. s 中有相邻两个相同字符。

值得注意的是，条件 2 保证了充要性。这里略去证明过程。

于是我们假设 s 经过若干次操作得到 t ，那么 s 必然可以被划分成最多 $|t| + 1$ 段，其中前 $|t|$ 段都能化作 t 的对应字符，并且可能有的第 $|t| + 1$ 段必须满足权值为 0。

也就是说， s 能经过若干次操作到 t 当且仅当能够成功按上述要求分段，且 s 存在相邻两个相同字符。证明过程比较繁琐，但是结论比较直观，有兴趣的同学可以自行推导。

于是，我们要模拟的就是 s 的这一分段过程，可以用一个简单的 dp 实现。令 $f(i)$ 表示 s 从 i 开始的后缀，能成功分段的 t 的数量，转移显然，具体可以参考代码。

时间复杂度 $O(T|s|)$ 。

3 神奇的花园 (garden.cpp/c/pas)

题解

如果没有强制在线，则将询问按 b_i 排序后依次添加路径，然后进行线段树合并进行统计即可。

强制在线则参考 Kruskal 算法，将所有边排序后进行线段树合并。每次合并两个连通块时新建一个点，将两个连通块的根的父亲设为这个新点，并在新点的点权记录当前这条边的长度。

询问的时候，就可以从 a_i 出发，向上倍增找到最远的点权符合 b_i 限制的那个点，然后读取线段树信息即可。时间复杂度 $O(n \log n + q \log n)$ 。

4 神奇的区间 (interval.cpp/c/pas)

题解

若两个区间 (a, b) 与 (c, d) 可相互到达，我们可以将其合并成一个大区间 (a, d) 。对于那些可以互达的点，可以通过并查集合并起来。由于所给出的区间的长度是递增的，所以对于一个区间 (a, b) ，我们在线段树中对应节点插入它。所有包含 a 或 b 的那些节点中的区间都可以与当前区间互达。对于不可互达的那些区间，他们之间的连通关系可以看成是一个森林（每棵子树都有根，且子节点可以走到父节点）。找出询问的两个区间的根节点然后直接判断即可。时间复杂度 $O(n \log n)$ 。

4.1 问题描述

小 A 穿越到了古老的区间国，这个国家的由一维数轴上的许多区间构成。

对于一维数轴上的两个区间 (a, b) 与 (c, d) ，若 $c < a < d$ 或 $c < b < d$ ，则可以从 (a, b) 单向走到 (c, d) 。区间国有两种操作：

1. 在区间集合中添加 (x, y) 区间，保证新加入的区间的长度一定比之前所有的区间长度要长；
2. 询问是否存在一条路径从第 x 个区间走到第 y 个区间。

初始时，区间集合为空。小 A 想探究区间国的形成，于是他给你发了一系列操作，想让你模拟一下区间国形成的过程，并回答所有的询问。

4.2 数据范围与约定

对于 30% 的数据， $1 \leq n \leq 10^3$ ；

对于 100% 的数据， $1 \leq n \leq 10^5$ ，所有数字绝对值 $\leq 10^9$ ，保证操作合法。