树上问题专题1-树上倍增和LCA-上

前置知识
讲解056、讲解057 - 并查集
讲解059 - 链式前向星建图
讲解078、讲解079 - 树型dp，系统学习动态规划看讲解066~讲解088
讲解117 - 倍增算法和ST表

树上问题专题讲述顺序
专题1：树上倍增和LCA-上，讲解118，本节
专题2：树上倍增和LCA-下，讲解119
专题3：树的重心，讲解120
专题4：树的直径，讲解121
专题5：树上差分，讲解122
专题6：换根dp，讲解123
树的静态点分治、树的动态点分治、树链剖分、基环树dp、启发式合并等内容会在【挺难】阶段讲述

左程云

树上问题专题1-树上倍增和LCA-上

树
每个节点只有一个头节点的图结构，就叫做树
如果所有节点形成一棵树，那么节点数为n，那么边的数量为n-1，也可能整体是森林结构
树上问题所指的树，包括一叉树、二叉树、多叉树

树上倍增
1）建立每个节点的深度表，deep数组
2）建立每个节点往上跳1步、跳2步、跳4步、跳8步..能到达的节点编号，ST表
3）给定任意节点i，可以快速查询，从i节点往上走的路径中位于第s层的节点编号

生成deep数组时间复杂度O(n)
生成ST表时间复杂度O(n * logn)
单次查询时间复杂度O(log n)

左程云

树上问题专题1-树上倍增和LCA-上

题目1
树节点的第K个祖先
树上有n个节点，编号0 ~ n-1，树的结构用parent数组代表
其中parent[i]是节点i的父节点，树的根节点是编号为0
树节点i的第k个祖先节点，是从节点i开始往上跳k步所来到的节点
实现TreeAncestor类
TreeAncestor(int n, int[] parent)：初始化
getKthAncestor(int i, int k)：返回节点i的第k个祖先节点，不存在返回-1
测试链接：https://leetcode.cn/problems/kth-ancestor-of-a-tree-node/

树上问题专题1-树上倍增和LCA-上

LCA问题
给定树上的任意两点a和b，如何快速查询出a和b的最低公共祖先，常见的有三个方法
1）树上倍增
2）tarjan算法+并查集
3）树链剖分，后续的课会讲述！

左程云

树上问题专题1-树上倍增和LCA-上

题目2
树上倍增解决LCA问题
测试链接：https://www.luogu.com.cn/problem/P3379

算法过程：
1）先让a和b往上跳到同一层
2）利用倍增算法找到a和b的最低公共祖先
课上重点图解过程

如果节点数为n，建立预处理结构的时间复杂度O(n*logn)，单次查询的时间复杂度O(log n)

优势是可以在线查询，如果一共m条查询，那么查询的总复杂度O(m * logn)

树上问题专题1-树上倍增和LCA-上

树上倍增解决LCA问题，迭代版代码

如果数据量很大，用递归函数可能会爆栈，导致无法通过测试，此时要进行递归改迭代的工作

课上重点图解

左程云

树上问题专题1-树上倍增和LCA-上

罗伯特·塔扬(Robert Tarjan)
1986年图灵奖得主
1，发明了tarjan算法高效的解决lca问题
2，发明了强联通分量的高效求法
3，发明了点双联通分量的高效求法
4，发明了splay树
5，发明了斐波那契堆
6，参与了并查集的改进和证明工作
7，还有很多我看不懂的贡献

这是我偶像 ->

左程云

树上问题专题1-树上倍增和LCA-上

题目3
tarjan算法解决LCA问题
测试链接：https://www.luogu.com.cn/problem/P3379

算法过程：
1）处理所有问题，建好每个节点的问题列表，然后遍历树
2）来到当前节点cur，令visited[cur] = true，表示当前节点已经访问
3）遍历cur的所有子树，每棵子树遍历完都和cur节点合并成一个集合，集合设置cur做头节点
4）遍历完所有子树后，处理关于cur节点的每一条查询(cur, x)
　　如果发现x已经访问过，cur和x的最低公共祖先 = x所在集合的头节点
　　如果发现x没有访问过，那么当前查询先不处理，等到x节点时再去处理查询(x, cur)得到答案
课上重点图解过程

如果节点数n，查询数量m，过程总的时间复杂度O(n + m)，但是只能做批量离线查询

树上问题专题1-树上倍增和LCA-上

tarjan算法解决LCA问题，迭代版代码

如果数据量很大，用递归函数可能会爆栈，导致无法通过测试，此时要进行递归改迭代的工作

课上重点图解

左程云