翻转表

flowchart TD

    Start(开始) --> id1[/判断链表是否存在/]

    id1[/判断链表是否存在/]--> |不存在| ERROR(返回ERROR)

    id1[/判断链表是否存在/] --> |存在| id2[/判断链表长度/]

    id2[/判断链表长度/] --> |长度为0或1| OK(返回OK)-->Stop

    id2[/判断链表长度/]--> |长度大于1| 初始化指针

    初始化指针 --> 进入循环

    进入循环 --> id3[/判断是否为第一个元素/]

    id3[/判断是否为第一个元素/] --> |是| 指向NULL

    id3[/判断是否为第一个元素/]--> |否| 改变指针方向

    改变指针方向 --> 移动到下一个元素

    移动到下一个元素 --> id4[/判断是否到达链表末尾/]

    id4[/判断是否到达链表末尾/] --> |是| id5(返回反转后的链表)

    id4[/判断是否到达链表末尾/] --> |否| 继续循环-->改变指针方向

    ERROR-->Stop(结束)

    id5-->Stop

删除倒数元素

flowchart TD

    Start(开始) --> id1[/判断链表是否为空/]

    id1[/判断链表是否为空/] --> |是| ERROR(返回ERROR)

    id1[/判断链表是否为空/] --> |否| 遍历链表获取长度

    遍历链表获取长度 --> id2[/判断n是否超过长度/]

    id2[/判断n是否超过长度/]--> |是| ERROR(返回ERROR)

    id2[/判断n是否超过长度/]--> |否| 定位倒数第n个节点

    定位倒数第n个节点 --> 删除该节点

    删除该节点 --> return(返回OK)-->S

    ERROR-->S(结束)

排序表

flowchart TD

    Start((开始)) --> id1[/判断链表是否为空/]

    id1 --> |是| ERROR(返回ERROR)

    id1 --> |否| 获取链表长度

    获取链表长度 --> 进入外层循环

    进入外层循环 --> 初始化指针p和q

    初始化指针p和q --> 进入内层循环

    进入内层循环 --> id2[/判断p和q指向的元素大小/]

    id2 --> |p > q| 交换p和q指向的元素

    id2 --> |p <= q| 不交换元素

    交换p和q指向的元素 --> 移动指针p和q

    不交换元素 --> 移动指针p和q

    移动指针p和q --> id3[/判断内层循环是否结束/]

    id3 --> |是| 内层循环结束

    id3 --> |否| 继续内层循环

    内层循环结束 --> 判断外层循环是否结束

    判断外层循环是否结束 --> |是| 外层循环结束-->Stop

    判断外层循环是否结束 --> |否| 继续外层循环

    ERROR --> Stop(结束)

最近公共祖先

flowchart TD

    Start(开始) --> id1[/判断二叉树是否为空/]

    id1 --> |是| NULL(返回NULL)-->stop(结束)

    id1 --> |否| id2[/判断当前节点是否为e1或e2/]

    id2 --> |是| 返回当前节点

    id2 --> |否| 在左子树中查找

    在左子树中查找 --> 在右子树中查找

    在右子树中查找 --> id3[/判断e1和e2是否分别在左右子树中/]

    id3 --> |是| 返回当前节点-->stop(结束)

    id3 --> |否| 返回不为空的子树-->stop(结束)

翻转树

flowchart TD

    Start(开始) --> id[/判断二叉树是否为空/]

    id --> |是| OK(返回OK)-->stop(结束)

    id --> |否| 交换左右子树

    交换左右子树 --> 递归交换左子树

    递归交换左子树 --> 递归交换右子树

    递归交换右子树 --> 返回OK-->stop(结束)

最大路径和

flowchart TD

    Start(开始) --> id2[/判断二叉树是否为空/]

    id2 --> |是| ERROR(返回ERROR)-->stop(结束)

    id2 --> |否| 计算左子树的最大路径和

    计算左子树的最大路径和 --> 计算右子树的最大路径和

    计算右子树的最大路径和 --> 计算包含当前节点的最大路径和

    计算包含当前节点的最大路径和 --> 返回不包含当前节点的最大路径和

    返回不包含当前节点的最大路径和 --> id[/判断左右子树的最大路径和/]

    id --> |左子树最大路径和大于右子树| 返回左子树的最大路径和加上当前节点的值-->stop(结束)

    id --> |右子树最大路径和大于左子树| 返回右子树的最大路径和加上当前节点的值-->stop(结束)

文件保存

flowchart TD

    Start((开始)) --> 打开文件

    打开文件 --> 判断文件是否打开成功

    判断文件是否打开成功 --> |是| 获取线性表长度和最大表长

    获取线性表长度和最大表长 --> 将元素依次写入文件

    将元素依次写入文件 --> 关闭文件

    关闭文件 --> 返回OK

    判断文件是否打开成功 --> |否| 返回ERROR

    返回ERROR --> 结束

文件读取

flowchart TD

    Start(开始) --> 打开文件

    打开文件 --> id1[/判断文件是否打开成功/]

    id1 --> |是| 获取线性表长度和最大表长

    获取线性表长度和最大表长 --> 读取第一个元素

    读取第一个元素 --> id2[/判断是否读取完所有元素/]

    id2 --> |是| 关闭文件--> stop(结束)

    id2 --> |否| 将元素插入线性表

    将元素插入线性表 --> 读取下一个元素

    读取下一个元素 -->id2

    id1 --> |否| 返回ERROR

返回ERROR --> stop(结束)

多表管理

flowchart TD

    Start(开始) --> 输入k-->id2[/判断k范围是否合法/]-->|是|LIST

    subgraph LIST

       direction TB

    id1[/选择在第k个树上操作/]-.->k[(树a1)]

    id1[/选择在第k个树上操作/]-.->a[(树...)]

    id1[/选择在第k个树上操作/]-->b[(树k)]

    id1[/选择在第k个树上操作/]-.->c[(树...)]

    id1[/选择在第k个树上操作/]-.->d[(树MAX\_NUM)]

    end

LIST-->stop(结束)

id2[/判断k范围是否合法/]-->|否|返回ERROR

返回ERROR --> stop(结束)