题目：反转单链表，并且使用固定的内存空间

举例：1 2 3 4 5 6 7 8 9 ------>9 8 7 6 5 4 3 2 1

解决方法：

2 1 3 4 5 6 7 8 9

3 2 1 4 5 6 7 8 9

4 3 2 1 5 6 7 8 9

5 4 3 2 1 6 7 8 9

。。。。。。

。。。。。。

9 8 7 6 5 4 3 2 1

题目：给定一个n节点的二叉树，写出一个O(n)时间的非递归过程，将该数每个节点的关键字输出。要求除该树本身的存储空间外只能使用固定量的额外存储空间，且在过程中不得修改该树，即使是暂时的修改也不行。

中序遍历，先序遍历，后序遍历

红黑树

树的fixup，调整红色的节点下面没有红色子节点----【条件1】

X为红色节点，x.p为红色节点

设新加入的节点为x，其父节点为x.p，（考虑情况x的父节点为其父节点的父节点的左子节点），y=x.p.p.right

1. y为红色节点。那么，使x.p.p变为红色节点，x.p和y为黑色节点，然后再检查x.p.p是否满足【条件1】。若不满足，若任然是情况1，则继续
2. 否则，y为黑色节点，那么，若x为x.p的左子节点，先对x进行右旋，再对x进行右旋
3. 否则，对x的父节点进行右旋。

红黑树删除

1. 是根节点，直接用后继代替。
2. 若右子树为空，直接用前驱代替。
3. 若右子树不为空，用后继代替。
4. 对以上改变，判断是否破坏红黑树性质

红黑树五大性质：

1. 树的节点要么是黑色，要么是红色
2. 树的根节点是黑色
3. 树的叶节点是黑色（NIL）
4. 红色节点的子节点是黑色
5. 从一个节点出发到其所有叶节点的简单路径所包含的黑色节点个数相同（这保证了红黑树的平衡性）