二叉查找树BST：左子树结点小于根结点，右子树结点大于根结点。

BST的中序遍历得到的是一个有序的数组。

lintcode 86，二叉查找树的中序遍历，非递归实现的要点，首先肯定需要用到stack，之前递归实现时，其实是操作系统的栈在起作用；其次在next()查找下一个结点nxt时，定义一个current结点，初始状态其指向root，在next()中，当前结点的左子树如果非空，则将其最左边的结点以及沿途的结点都压栈，这样栈顶元素就是要找的nxt，这时的栈中不仅有nxt，还有一路通过寻找左孩子走到nxt时，所访问的所有结点，这样做的原因返回nxt的同时要将current结点指向其右孩子，再次调用next()时，如果current.right为空，要找的结点就是当前栈顶，因为它是以current为左子树结点的最接近current的结点，这样程序比较容易理解。