**难度: 简单**

在经典汉诺塔问题中，有 3 根柱子及 N 个不同大小的穿孔圆盘，盘子可以滑入任意一根柱子。一开始，所有盘子自上而下按升序依次套在第一根柱子上(即每一个盘子只能放在更大的盘子上面)。移动圆盘时受到以下限制:

(1) 每次只能移动一个盘子;

(2) 盘子只能从柱子顶端滑出移到下一根柱子;

(3) 盘子只能叠在比它大的盘子上。

请编写程序，用栈将所有盘子从第一根柱子移到最后一根柱子。

你需要原地修改栈。

示例1:

输入：A = [2, 1, 0], B = [], C = []

输出：C = [2, 1, 0]

示例2:

输入：A = [1, 0], B = [], C = []

输出：C = [1, 0]

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/hanota-lcci

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

思路:

1. 将A借助C移动到B

2. 将A最下面的一个移动到C

3. 将B借助A移动到C

|  |
| --- |
| /\*\*  \*  \* @param A 原柱子  \* @param B 辅助柱子  \* @param C 目标柱子  \*/  public void hanota(List<Integer> A, List<Integer> B, List<Integer> C) {  movePlate(A.size(),A,B,C);  }  private void movePlate(int num, List<Integer> A, List<Integer> B, List<Integer> C) {  if (num == 1) {  //如果只有一个,直接把A移动到C  C.add(A.remove(A.size() - 1));  return;  }  //将A借助C移动到B  movePlate(num - 1,A,C,B);  //将A的最下面的一个移动到C  C.add( A.remove( A.size() - 1));  //将B借助A移动到C  movePlate(num - 1,B,A ,C);  } |