

Assignment #1

1. Problems: Tower of Hanoi

You will implement a solver for the Tower of Hanoi problem. Tower of Hanoi puzzle has three bars and n disks of different sizes. Let the three bars be numbered 1, 2, and 3. N disks are stacked on bar 1 in ascending order of the sizes with the largest disk at the bottom and smallest disk at the top as Figure 1. Your goal is to move this entire disks from bar 1 to bar 3 with a minimal number of movements. **You should obey the following rules:**

- A. Only one disk can be moved at a time.
- B. Each move consists of taking the upper disk from one of the stacks and placing it on top of the other stack or empty bar.
- C. A larger disk cannot be placed on a smaller disk.

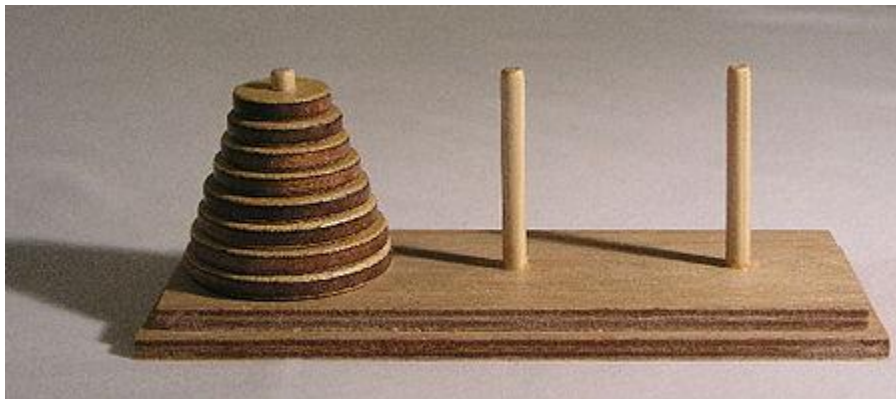


Figure 1: Tower of Hanoi with eight disks

2. Implement Tower of Hanoi with 3 bars

- **Use recursion to solve the Tower of Hanoi problem.**
- **num_disks:** Number of disks in bar 1.
- **start_pos:** 1, 2, or 3 denoting the number of bars on which stack of disks is initially located.
- **end_pos:** 1, 2, or 3 denoting the number of bars on which stack of disks is located at the end.

- **Return value:** the optimal number of movements to solve the Tower of Hanoi problem; when num_disk is three, you should return seven.
- Please count each movement of disks using recursion.

3. Add movement log

Add disk movement logging code to hanoi solver.

You should use *System.out.println()* to print each disk movement using the format below.

Output format example (when num_disks=3, from_bar=1, second_bar=2, to_bar=3)

Disk 1 moved from bar 1 to bar 3
Disk 2 moved from bar 1 to bar 2
Disk 1 moved from bar 3 to bar 2
Disk 3 moved from bar 1 to bar 3
Disk 1 moved from bar 2 to bar 1
Disk 2 moved from bar 2 to bar 3
Disk 1 moved from bar 1 to bar 3