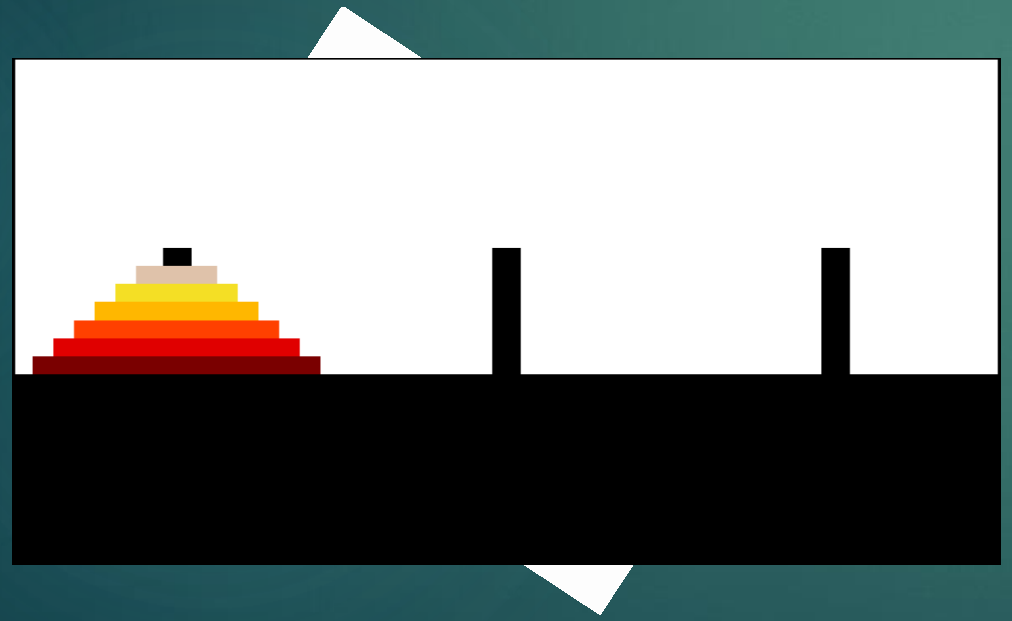
The Peg Puzzle Peril – A Physical Wooden Stick Puzzle Game

Reference: Manish Chandra Joshi | Reg. No: 12302359

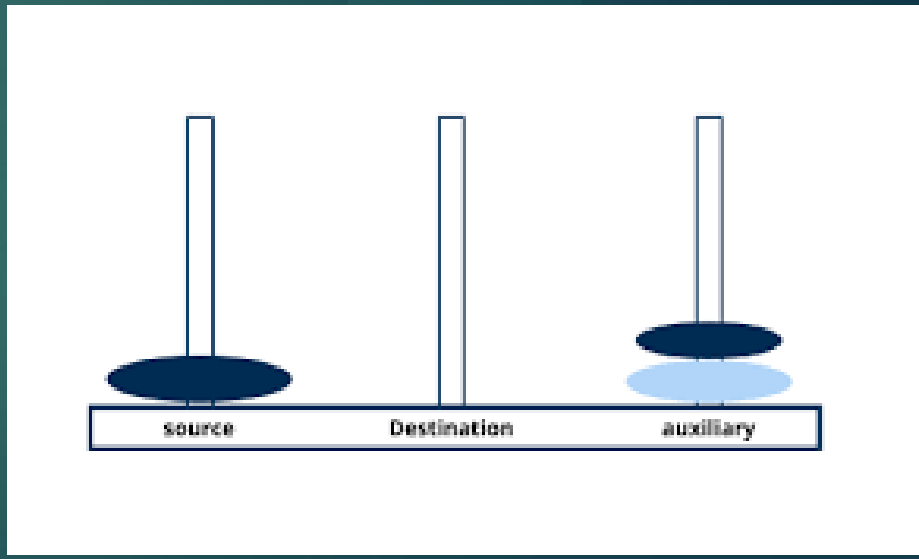


Rules & Constraints

- The game consists of **three vertical wooden sticks**.
- A **mountain-like structure**, divided into **3-4 separate wooden blocks**, is **stacked on the first stick** (largest at the bottom, smallest at the top).
- The **goal** is to move the mountain **from the first stick to the third stick** while following the rules.

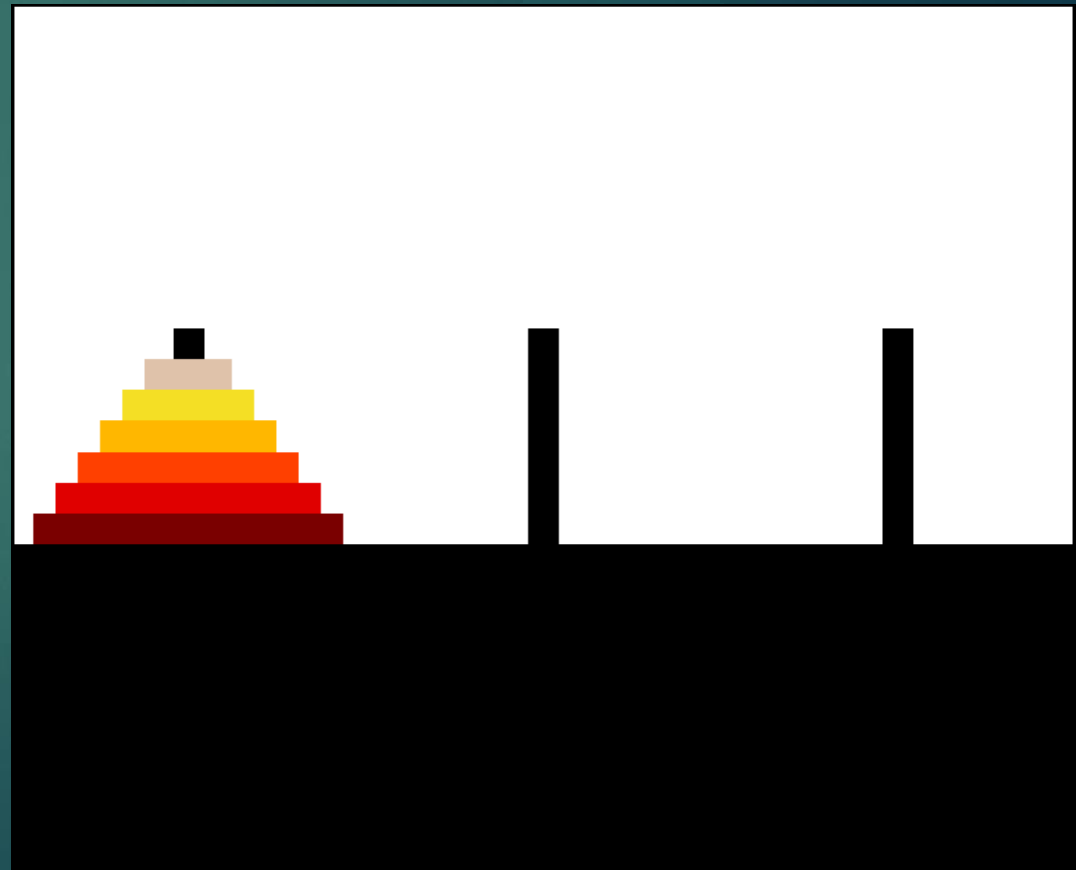
1-MOVE ONE BLOCK AT A TIME.
2-LARGER BLOCK CANNOT BE PLACED ON A SMALLER BLOCK.
3-USE THE SECOND STICK AS A HELPER (OPTIONAL).
4-WINNING CONDITION.
5-LOSING CONDITION.
6-SCORING & GAMIFICATION (OPTIONAL).

- **MINIMUM MOVES CHALLENGE:** PLAYERS MUST TRY TO SOLVE THE PUZZLE IN THE **FEWEST POSSIBLE MOVES.**
- **TIME-BASED CHALLENGE:** PLAYERS CAN HAVE A **TIME LIMIT** TO COMPLETE THE PUZZLE.
- **MULTIPLE DIFFICULTY LEVELS:** MORE BLOCKS = **HIGHER DIFFICULTY**.

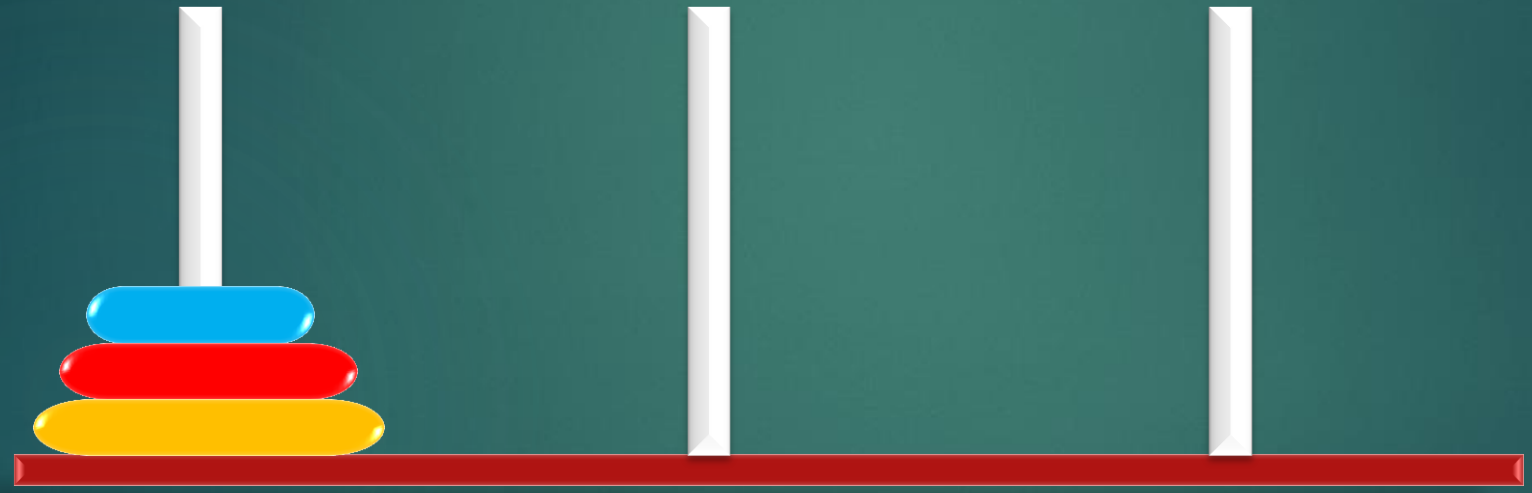


Animated solution with winner's cup

- **Step-by-Step Solution**
- 1. Move **Disk 1** from **A** → **C**
  2. Move **Disk 2** from **A** → **B**
  3. Move **Disk 1** from **C** → **B**
  4. Move **Disk 3** from **A** → **C**
  5. Move **Disk 1** from **B** → **A**
  6. Move **Disk 2** from **B** → **C**
  7. Move **Disk 1** from **A** → **C**



INITIAL POSITION

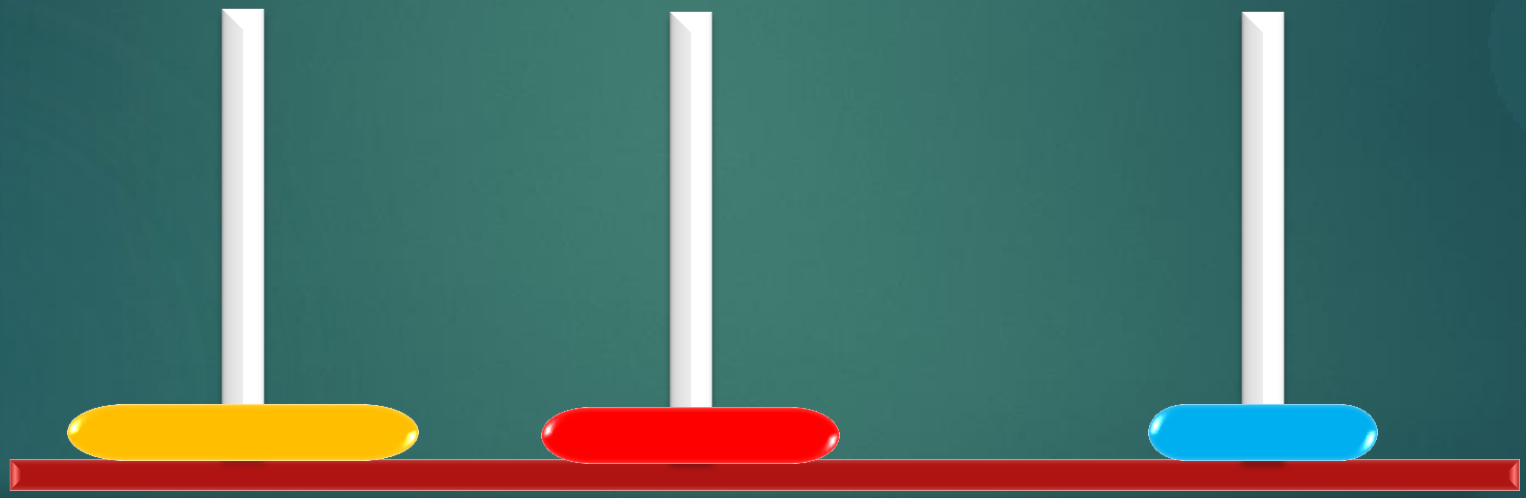


Move 1

1 Move **Disk 1** from **A** → **C**

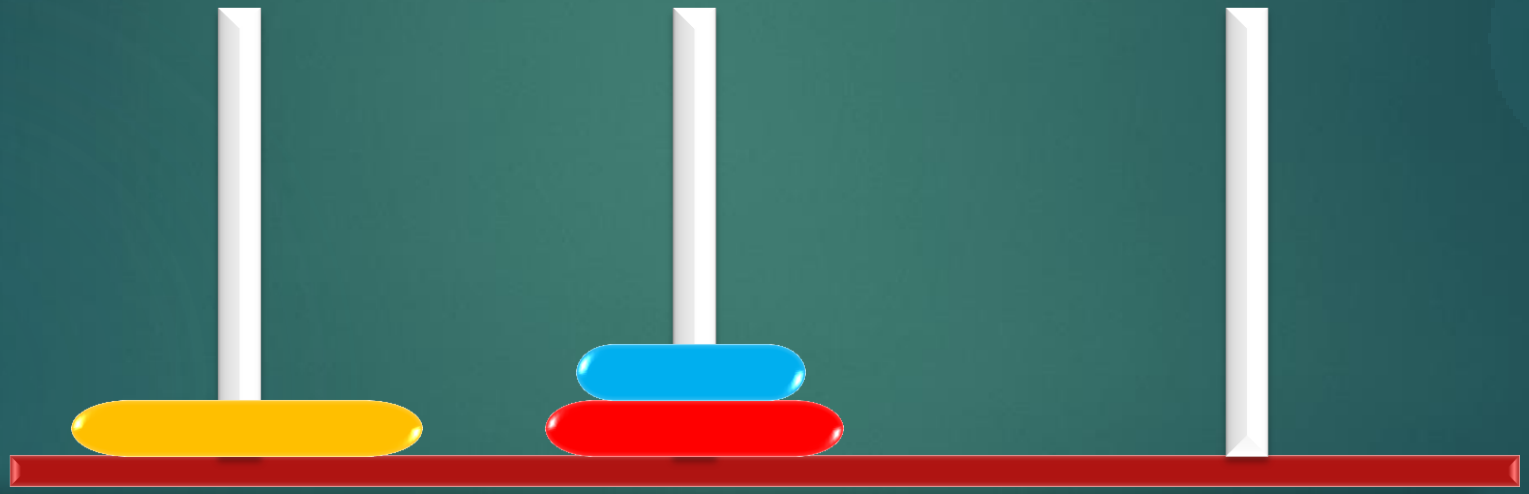
Move 2

2 Move **Disk 2** from **A** → **B**



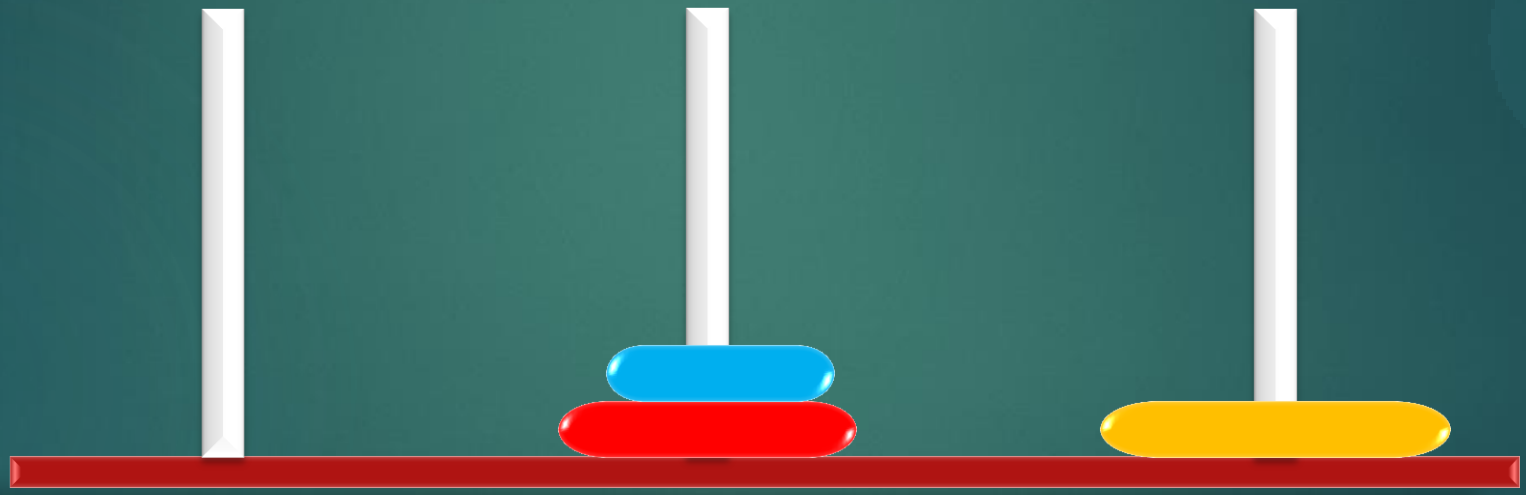
Move 3

3 Move **Disk 1** from **C** → **B**



Move 4

4. Move **Disk 3** from **A** → **C**

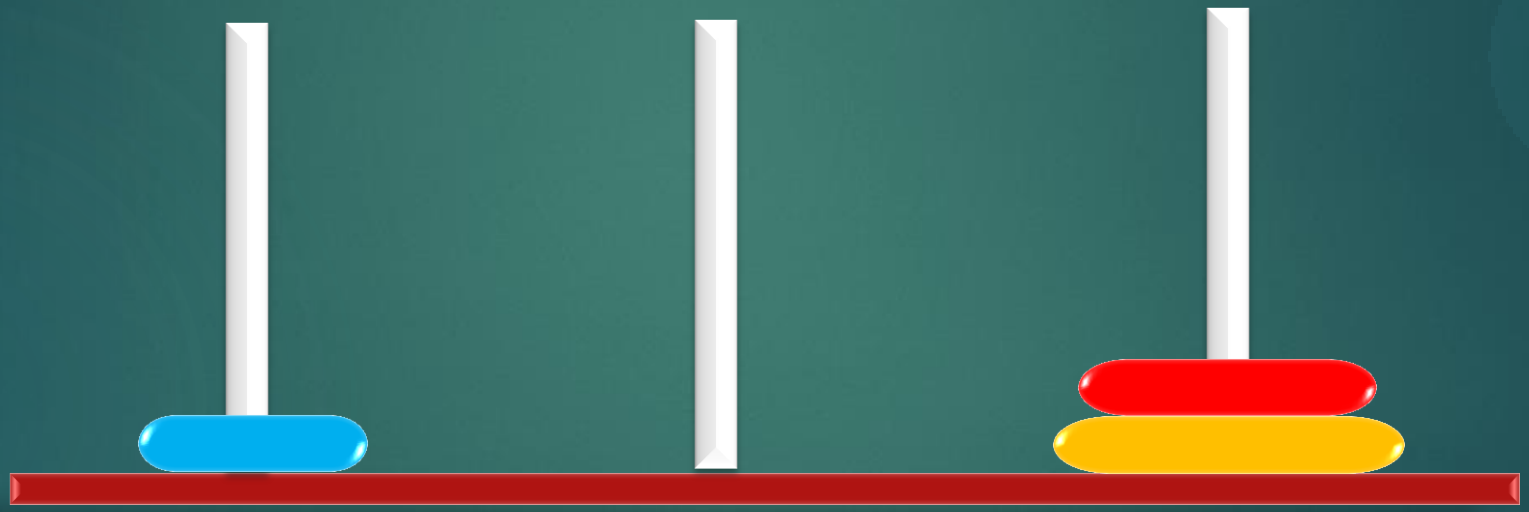


Move 5

5 Move **Disk 1** from **B** → **A**

Move 6

6 Move **Disk 2** from **B** → **C**



Move 7

7 Move **Disk 1** from **A** → **C**

Finally, **after 7moves**, all 3disks will be stacked in the correct order on **Peg 3**.

CODE-

CA 1.pl [modified]

```
puzzle(0, _, _, _) :- !.  % Base case: No move when there are 0 disks.

puzzle(N, Source, Auxiliary, Destination) :-
    N > 0,
    M is N - 1,

    % Move N-1 disks from Source to Auxiliary using Destination
    puzzle(M, Source, Destination, Auxiliary),

    % Move the largest disk from Source to Destination
    write('Move disk '), write(N), write(' from '), write(Source), write(' to '), wr
ite(Destination), nl,

    % Move N-1 disks from Auxiliary to Destination using Source
    puzzle(M, Auxiliary, Source, Destination).

% Run this to solve for 3 disks
solve_puzzle :- puzzle(3, 'A', 'B', 'C').
```

OUTPUT-

```
% c:/users/manish/onedrive/documents/prolog/ca 1 compiled 0.02 sec, -2 clauses
?- solve_puzzle.
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
true.

?-
```