

# Tower of Hanoi

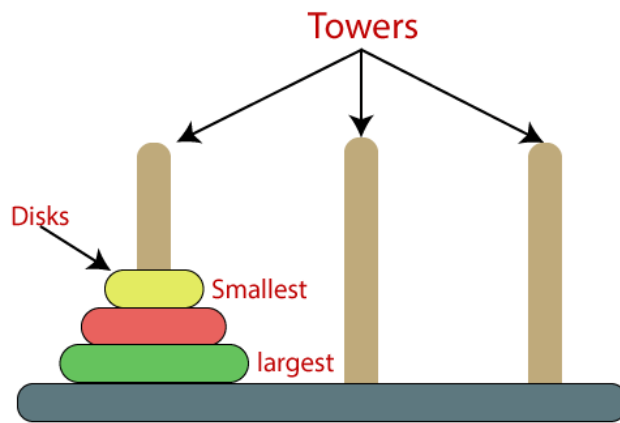
**Objective:** To create a puzzle solver using recursion.

**Background:**

The Tower of Hanoi puzzle, an intricate mathematical conundrum, was conceived by the French mathematician Édouard Lucas and initially presented as a game in a document published in 1883. In that document it supposed a fantasy that the puzzle originated from a legend known as the “Sacred Tower of Brahma,” which comprised 64 golden disks. The legend stated that the completion of the tower would herald the end of the world. Assuming that a single disk is moved every second, the tower would be completed in approximately 585 billion years, which is approximately 42 times the age of the universe.

The Tower of Hanoi game consists of three rods anchored to a board and a number of disks of various diameters that can be moved from rod to rod. The enduring popularity of the puzzle has spawned various forms, including online versions.

The game begins with the disks stacked on the left rod (or tower) in order of decreasing size. The objective is to move the whole stack from the left rod to the middle rod and have them stacked in the same order.

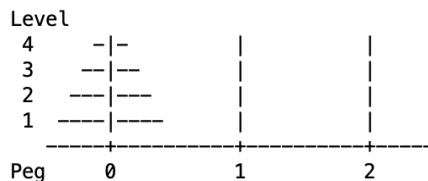


The movement of the disks follow these simple rules.

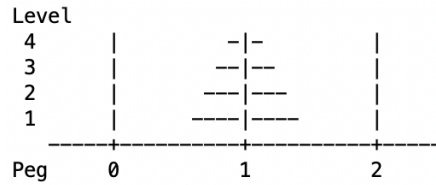
1. Only one disk can be moved at a time.
2. Only a top disk can be moved.
3. No disk can be placed on top of a smaller disk.

If done properly, the minimum number of moves required for  $n$  disks is  $2^n - 1$ .

You will be provided a working version of the game in which you specify the number of disks and you make the moves. A game with 4 disks starts like this:



The game only allows you to move disks according to the rules. You complete the game when all disks are in the center (tower 1).



The game will declare “SUCCESS!!! You won!” and report how many moves it took.

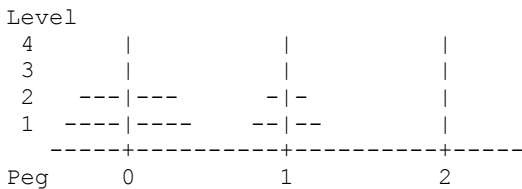
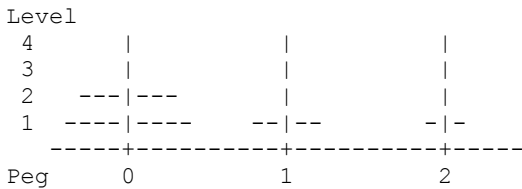
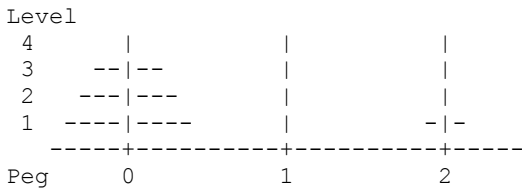
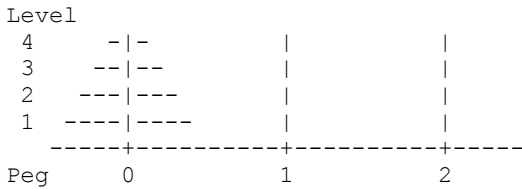
**Assignment:**

Create the directory **HanoiTower** and do all of your work in that directory. Go to Mr Greenstein’s web site and copy the code for HanoiTower into a Geany window and save as **HanoiTower.java**.

1. Compile and play the game several times. Come up with a recursive strategy to have the program solve the game in the least number of moves.
2. In the run method, comment out **playGame(args)** and uncomment out **solvePuzzle(args)**. Create a **solvePuzzle** method that solves the number of levels recursively. Have it print out each move to the screen.

A sample run of **solvePuzzle**:

```
% java HanoiTower 4
```



-----+-----+-----+-----  
Peg      0            1            2

Level

4        |            |            |  
3        |            |            |  
2        -|-           |            |  
1        -|-|-        -|-|-        -|-|-  
-----+-----+-----+-----  
Peg      0            1            2

Level

4        |            |            |  
3        |            |            |  
2        -|-           |            -|-|-  
1        -|-|-        |            -|-|-  
-----+-----+-----+-----  
Peg      0            1            2

Level

4        |            |            |  
3        |            |            -|-  
2        |            |            -|-|-  
1        -|-|-        |            -|-|-  
-----+-----+-----+-----  
Peg      0            1            2

Level

4        |            |            |  
3        |            |            -|-  
2        |            |            -|-|-  
1        |            -|-|-        -|-|-  
-----+-----+-----+-----  
Peg      0            1            2

Level

4        |            |            |  
3        |            |            |  
2        |            -|-           -|-|-  
1        |            -|-|-        -|-|-  
-----+-----+-----+-----  
Peg      0            1            2

Level

4        |            |            |  
3        |            |            |  
2        |            -|-           |  
1        -|-|-        -|-|-        -|-|-  
-----+-----+-----+-----  
Peg      0            1            2

Level

4        |            |            |  
3        |            |            |  
2        -|-           |            |  
1        -|-|-        -|-|-        -|-|-  
-----+-----+-----+-----  
Peg      0            1            2

Level

4        |            |            |  
3        |            |            |  
2        -|-           -|-|-        |  
1        -|-|-        -|-|-        |

```

-----+-----+-----+-----
Peg      0         1         2

```

Level

```

4         |         |         |
3         |         |         |
2         |         ---|---   |
1        --|--     ---|---   -|-
-----+-----+-----+-----
Peg      0         1         2

```

Level

```

4         |         |         |
3         |         --|--     |
2         |         ---|---   |
1         |         ---|---   -|-
-----+-----+-----+-----
Peg      0         1         2

```

Level

```

4         |         -|-       |
3         |         --|--     |
2         |         ---|---   |
1         |         ---|---   |
-----+-----+-----+-----
Peg      0         1         2

```

It took you 15 moves with 4 levels.