

Round 3 -Lab Manual

Name of Faculty: Mr.Praveen Goyal

Institute: Rajkiya Engineering College,Banda

Email ID : praveen_soit@yahoo.com

Discipline to which the Lab belongs: Information Technology

Name of the Lab: Data Structure

Name of experiment: Tower of Hanoi

Kindly Refer these documents before filling the worksheet

1. Coursework (MOOC) on Pedagogy , Storyboard , Lab Manual : [h ttp://bit.ly/Vlabs-MOOC](http://bit.ly/Vlabs-MOOC)
2. Additional Documentation booklet for reference. <http://vlabs.iitb.ac.in/vlabs-dev/document.php>
3. Sample Git Repository.: <https://github.com/BootTeam11/Boot2k19.git>

Round 2

1. Aim and Objective

To understand and code the problem of Tower of Hanoi using recursion for n disks.

2. Theory

The Tower of Hanoi is a mathematical game or puzzle. It consists of three rods, and a number of disks of different sizes which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape. The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

- Only one disk can be moved at a time.

- Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.

- No disk may be placed on top of a smaller disk. With three disks, the puzzle can be solved in seven moves.

The minimum number of moves required to solve a Tower of Hanoi puzzle is $2^n - 1$, where n is the number of disks.


3. Procedure (Protocol for navigating through the simulator with screenshots)

b3.1 Follow the steps as mentioned above and below the webpage.

Tower of Hanoi


Understanding and Coding the Puzzle!

BEGIN




A Glimpse at the Puzzle

Glance through the history of the Puzzle and the know the instructions that make the Puzzle so captivating.




Game Play of 3 disks

Try to move the stack of 3 disks from first to last rod using spare rod too.




Game Play of 7 disks

Try to move the stack of 7 disks from first to last rod using spare rod too.



Code


Understand the Logic behind the Puzzle.



Quiz

Test your knowledge.


A Glimpse at the Puzzle



The puzzle was invented back in 19th century. It was played by young priests in Hindu Temples to furnish their mental discipline and focus. They practiced the puzzle with 50 disks arranged on one of the three plates. Each disk is placed on slightly larger disk. The aim is to re-create the tower formed to a different plate by moving disks one disk at a time. In addition to this condition, a larger disk cannot be placed on smaller disk. As we increase the number of disks we observe patterns which helps us decipher this conventional and celebrated puzzle. One may utilise the

Instructions

1. Only one disk can be moved at a time.
2. Each move consists of taking the upper disk from one of the stacks and placing on top of another stack. In other words, a disk can only be moved if it is the uppermost disk on a stack.
3. No larger disk may be placed on top of a smaller disk.



1

Play with 3 disks:
Start Easy

2

Play with 7 disks:
Observe the pattern when the
Puzzle becomes more tough.

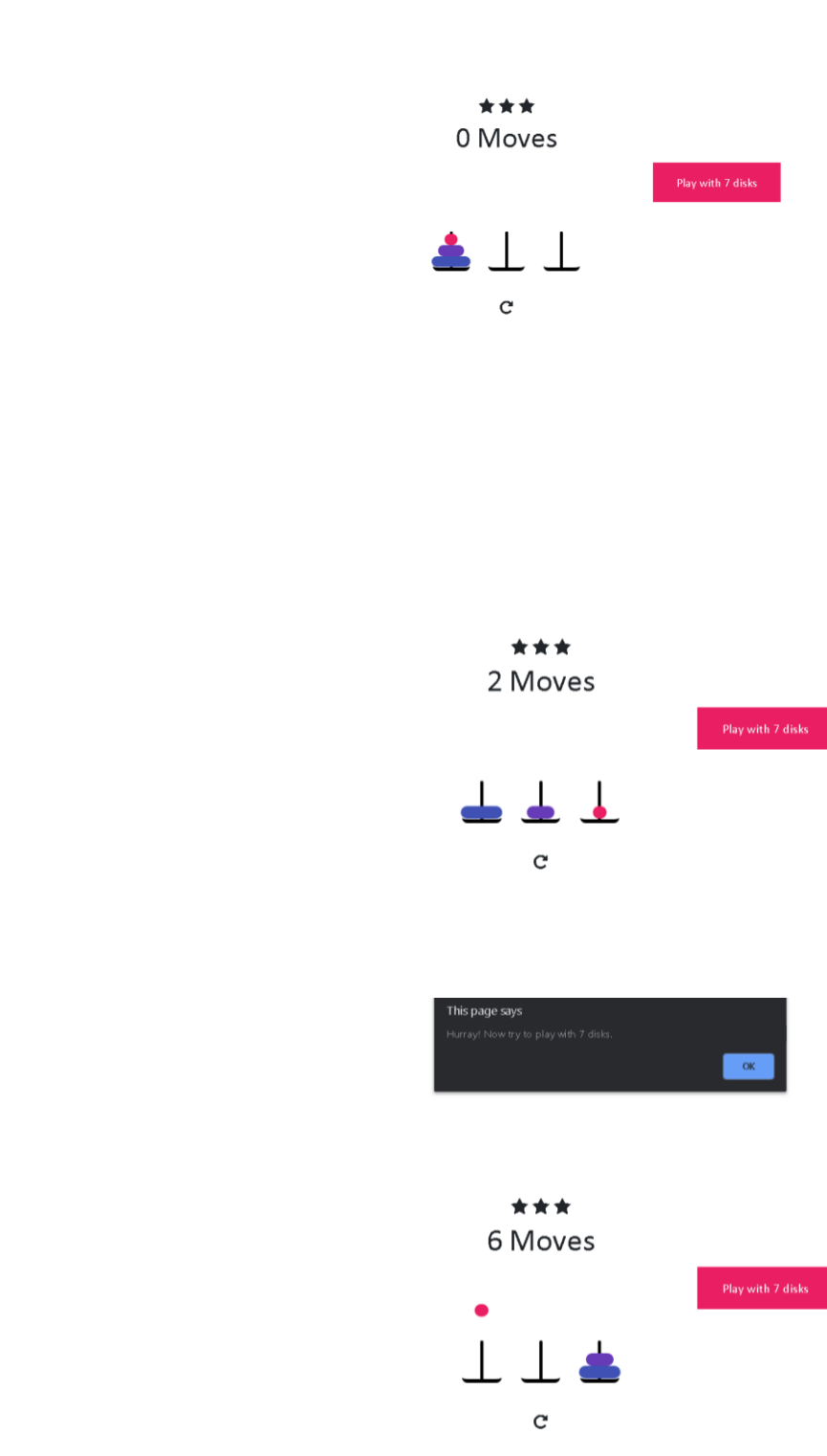
3

Code:
Understand the significant
concepts and solution.

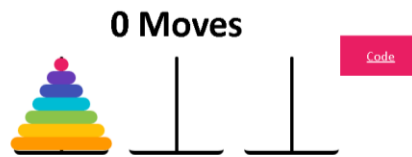
4

Quiz:
Test your Understanding.

3.2 Play the Puzzle with 3 disks:



3.3 Similarly play the Puzzle with 7 Disks:



3.4 Proceed to Code:

Logic and Reasoning

By now, you might have identified that to move N disks from one peg to another, you need $2N-1$. So, the number of steps almost double every time you insert another disk in the stack.

Let us prove that the number of steps in $2N-1$. The question is what is the minimum number of moves (a_N) required to move all the N -disks to another peg.

Let's look at a recursive solution.

One can already see that $a_1=1, a_2=3, a_3=7$ and so on. For a given N number of disks, the way to accomplish the task in a minimum number of steps is:

Move the top $N-1$ disks to an intermediate peg.

Move the bottom disk to the destination peg.

Finally, move the $N-1$ disks from the intermediate peg to the destination peg.

Therefore, the recurrence relation for this puzzle would become:

$$a_1=1, a_2=3, a_N=2a_{N-1}+1, N \geq 2$$

$$\begin{aligned} a_N &= 2a_{N-1} + 1 \\ &= 2(2a_{N-2} + 1) + 1 \\ &= 2^2a_{N-2} + 2 + 1 \\ &= 2^3a_{N-3} + 2^2 + 2 + 1 \\ &= 2^{N-1}a_{N-(N-1)} + 2^{N-2} + \dots + 2^2 + 2 + 1 \\ &= 2^N - 1 \end{aligned}$$

Recursion

In case of 3 disks,

Move Disk 1 from peg A to peg C, Then move disk 2 from peg A to peg B and, finally, move disk 1 from peg C to peg B.

This solution takes 3 steps.

You can easily move this stack from peg B to any other peg using these 3 steps.

But what if we have 3 disks - 1, 2, and 3 stacked in peg A.

To move the stack to peg B you would have to expose disk 3 and to do that disk 1 and 2 have to be moved to peg C.

So by ensuring that you do not break the rules, using the above subsets of 3 steps in the previous case, you would move disk 1 and 2 to peg C, leaving disk 3 exposed with no disk above it.

Now to solve the problem, recursively move disk 3 from peg A to peg B.

Then disk 1 from peg C to peg A. After which disk 2 can be moved above disk 3 at peg B.

The puzzle is finally completed by moving disk 1 from peg A over disk 2 and 3 at peg B.

What if you have 4 disks?

Algorithm

Create a function Tower with int 'a' - for number of disks, char 'from' - for from peg, char 'aux' - for a secondary peg, char 'to' - for destination peg

Put 'if' loop

If (a=1) i.e. if number of disk = 1, move it from 'initial peg' to the 'destination peg'

Else, call function tower for 'a-1' i.e. the number of disk -1, recall function tower for n-1 disk and move it 'from' to 'to' Recall function again using recursion until an or number of the disk is 1.

Code

```
function Towers_of_Hanoi (n, initial_peg, final_peg, temp_peg)

if n ==1
    S1      = 'Move one disk from ';
    S2      = num2str(initial_peg);
    S3      = ' to ';
    S4      = num2str(final_peg);
    message = [S1,S2,S3,S4];
    disp(message);
else
    % move n-1 disk from peg 1 to peg 2 and use peg 3 as a temporarily
    % holding peg.
    Towers_of_Hanoi ( n-1, initial_peg, temp_peg, final_peg )
    % move the last disk from peg 1 to peg 3.
    Towers_of_Hanoi ( 1, initial_peg, final_peg ,temp_peg)
    % move the n-1 disk from peg 2 to peg 3 using peg 1 as the temporarily
    % holding peg.
```

3.5 Proceed to Quiz Questions

4.Pre test Assessments (Highlight the correct option with bold text)

1. What is the name of the equation that is used to solve this puzzle?

- a. Geometric Progression
- b. Arithmetic Progression
- c. Fibonacci Sequence
- b. **Recurrence Relation (Correct answer)**

2.The Tower of Hanoi has many useful applications.It is used in which of the following fields :

- a.Computer Programming and Algorithms
- b.Psychological Research
- c.Data Backup rotation scheme
- d.All of these

5. Post test Assessments (*Write least one question for each learning objective from round 1*)

For Learning Objective 1

1. What is the number of moves required to solve Tower of Hanoi problem for k disks?
 - a. $2k-1$
 - b. $2k+1$
 - c. 2^k+1
 - d. 2^k-1

For Learning Objective 2

2. Tower of Hanoi problem can be solved iteratively
 - a. True
 - b. False
3. The optimal Data Structure used to solve Tower of Hanoi is:
 - a. Tree
 - b. Priority Queue
 - c. Heap
 - d. Stack

