IC250 Laboratory Assignment - 02

Aditya Nigam

August 21, 2016

IC250 Programming and Data Structure Practicum

Lab Assignment No. 02 Date : {23,24,25} Aug , 2016

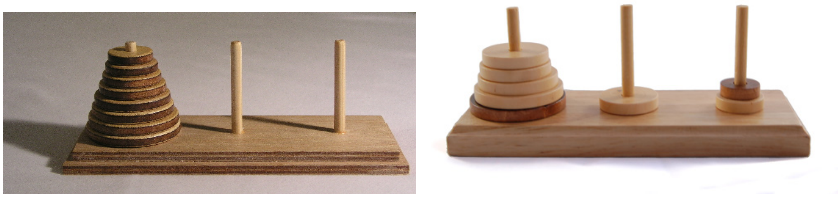
Introduction/Problem Context

This laboratory assignment focuses on use of stack data structure in the context of solving some general puzzles. This data structure can only allow to access a special element and that by its own specific routine. Also their insertion and deletions are also different.

It assumes that you are familiar with static and dynamic data representation and C language features related to them. You may refer to the references given, if you required to refresh these topics.

Problem : Tower of Hanoi

It consists of three rods, and a number of disks of different sizes which can slide onto any rod as shown in Figure 2. 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:



(a) Starting

(b) After Some Steps

Figure 1: Tower of Hanoi Setup

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 it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
3. 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.

Task Description

You are required to write a C program which take the number of disks (n) from user and output the set of moves to shift the entire stack to another rod, obeying the above defined rules.

Input Data and Format

Enter the number of disks n.

Computation Involved

1. Write the recursive solution.
2. Write the stack based iterative solution.

Pseudocode

The Pseudocode for recursive version looks like :

```
FUNCTION DiscMove(disk, source, dest, spare):
IF disk == 0, THEN:
    move disk from source to dest
ELSE:
    MoveTower(disk - 1, source, spare, dest)   // Step 1 above
    move disk from source to dest              // Step 2 above
    MoveTower(disk - 1, spare, dest, source)   // Step 3 above
END IF
```

This recursive code can also be realized in an iterative version. Recursive function use stack implicitly. In iterative version, stack has to be used explicitly to do all book keeping.

Expected Output for Correct and Incorrect Inputs

For any number of disks output the steps that are required to shift the entire stack in the following format.

1. Move Disk from **Peg A** $->$ **Peg B**
2. Move Disk from **Peg B** $->$ **Peg C**
3. Move Disk from **Peg A** $->$ **Peg C**
4. ...
5. ...

Sample Output

```
$ hanoi

Number of disks : 3

Disk movement sequence to solve it is as follows:

[1] Move disk 1 from peg A to peg C
[2] Move disk 2 from peg A to peg B
[3] Move disk 1 from peg C to peg B
[4] Move disk 3 from peg A to peg C
[5] Move disk 1 from peg B to peg A
[6] Move disk 2 from peg B to peg C
[7] Move disk 1 from peg A to peg C

Total number of steps required to solve this problem are 7.
```

Optional Part

- Estimate the number of steps requires to solve the puzzle in terms of number of disks n.
- **Double Tower of Hanoi :** The goal is to move a double tower of Hanoi, *i.e.* a tower of Hanoi that contains $2 \times n$ disks of n different sizes (2 of each size), from the first peg to the third in minimal number of moves.

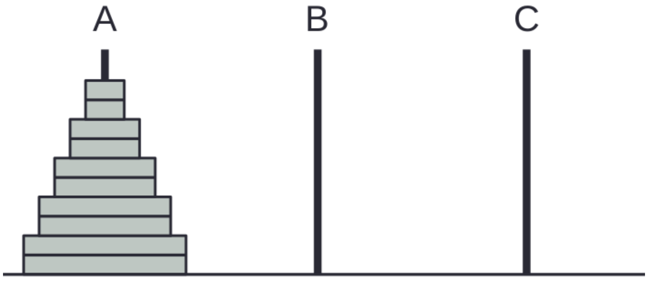


Figure 2: Double Tower of Hanoi Setup

References

- https://en.wikipedia.org/wiki/Tower_of_Hanoi
- http://www.cut-the-knot.org/recurrence/hanoi.shtml
- http://mathworld.wolfram.com/TowerofHanoi.html
- http://www.cs.cmu.edu/~cburch/survey/recurse/hanoiimpl.html