Lab 11

# Recursion

## Background

In this lab, you will be introduced to the concept of recursion. We will take a program using abstract stack and solve a popular puzzle.

## Printing a string

Implement the pseudo-code below in C++:

```
PRINT-STRING(string)
   if LENGTH(string) = 0: // base case
      return

   PRINT-CHAR(FIRST(s))    // do some work
   PRINT-STRING(REST(s))   // recursive call on simpler problem
   PRINT-CHAR(FIRST(s))    // do some work
```

Test the function using different strings and find out what the functions outputs.

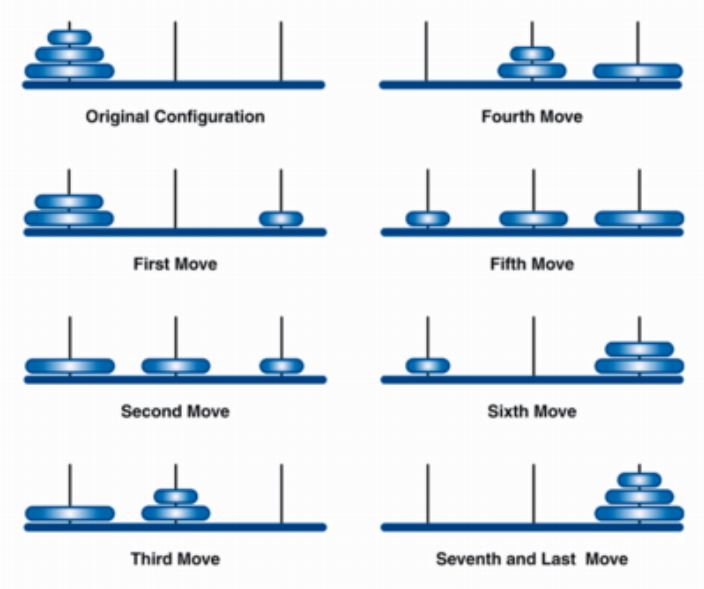
## Towers of Hanoi

Towers of Hanoi puzzle comes from history, monks in Vietnam were asked to carry 64 gold disks from one tower (stack) to another. How will the monks solve this problem? How long will it take them?

The puzzle consists of three towers (or pegs) with a specific number of disks with different sizes in one tower, arranged in ascending order (from the top) from smallest to largest. The goal is to move the stack of disks from the source tower to the destination tower. But there are certain rules to be followed.

The rules of the puzzle are:

- Only one disk may be moved at a time.
- Each move consists of taking the upper disk from one of the towers and sliding it onto another tower, on top of the other disks that may already be present on that tower.




- No disk may be placed on top of a smaller disk.

We need to accomplish this by using the least possible number of moves in each number of disks. The least number of moves is: $2^n - 1$ where $n$ is the number of disks. This will give us the time needed by the monks to complete the puzzle with 64 disks, once completed, it is believed to be the end the world.

## The recursive algorithm

The easiest solution is a recursive one. The key to the solution is to notice that to move any disk, we must first move the smaller disks off of it, thus a recursive definition. Another way to look at it is this, if we had a function to move the top three disks to the middle position, we could put the biggest disk in its place. All we need to do is assume we have this function and then call it.

- Lets start with 1 disk (our base case): Move 1 disk from source tower to destination tower and we are done.
- To move 2 disks: Move smaller disk from source tower to spare tower, move larger disk from source tower to destination tower, move smaller disk from spare tower to destination tower and we are done.
- To move n disks (or think of, say, 3 disks): Solve the problem for $n-1$ disks (i.e. 2 disks) using the spare tower instead of the destination tower (i.e. get 2 disks onto the spare tower). Then, move the biggest disk from source tower to destination tower. Then again solve the problem for $n-1$ disks but use the spare tower instead of the source tower (i.e. get the 2 disks onto the destination tower using the source tower as the spare tower).

Since the upper disks are moved from the towers, which means Last In, First Out data structure—the stack—perfect fits the puzzle. We create three stacks for the three towers: source, destination, and spare. Initially, n disks are placed onto the source stack with the smallest numbered 1 and the largest numbered n. The algorithm will be as following:

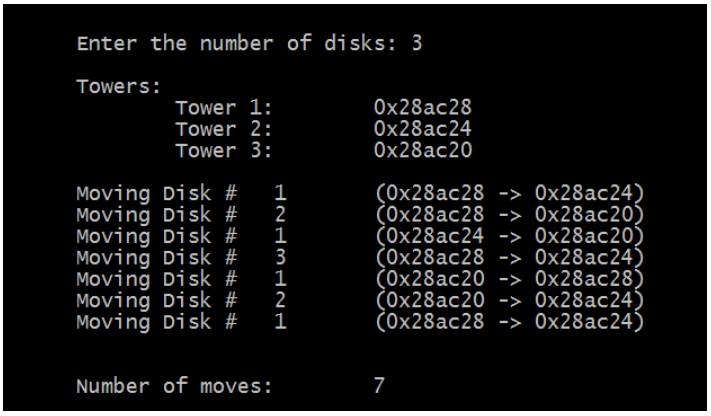



**Figure 11.1:** Output on console for the task.

```
MOVE-TOWERS-OF-HANOI(disk, source, destination, spare):
   if disk == 1:
      MOVE-DISK(source, destination)
   else:
      MOVE-TOWERS-OF-HANOI(disk - 1, source, spare, destination)
      MOVE-DISK(source, destination)
      MOVE-TOWERS-OF-HANOI(disk - 1, spare, destination, source)
```

Note that the pseudocode adds a base case: When disk is 1, the smallest disk. In this case we don't need to worry about smaller disks, so we can just move the disk directly. In the other cases, we follow the three-step recursive procedure.

## Hand in

Hand in the source code from this lab at the appropriate location on the blackboard system at LMS. You should hand in a single compressed/archived file that contains the following.

1. C++ source code files representing the work accomplished for this lab. All source code files should contain author in the comments at the top of the file. It is expected that you will have a file for the task (Lab_5_-_Towers_of_Hanoi.cpp) with the output shown in Figure 11.1.

2. A plain text file named OUTPUT.txt that includes a) author information at the beginning, b) a brief explanation of the lab and c) shows the compile and run steps of your code. The best way to generate this file is to cut and paste from the command line.