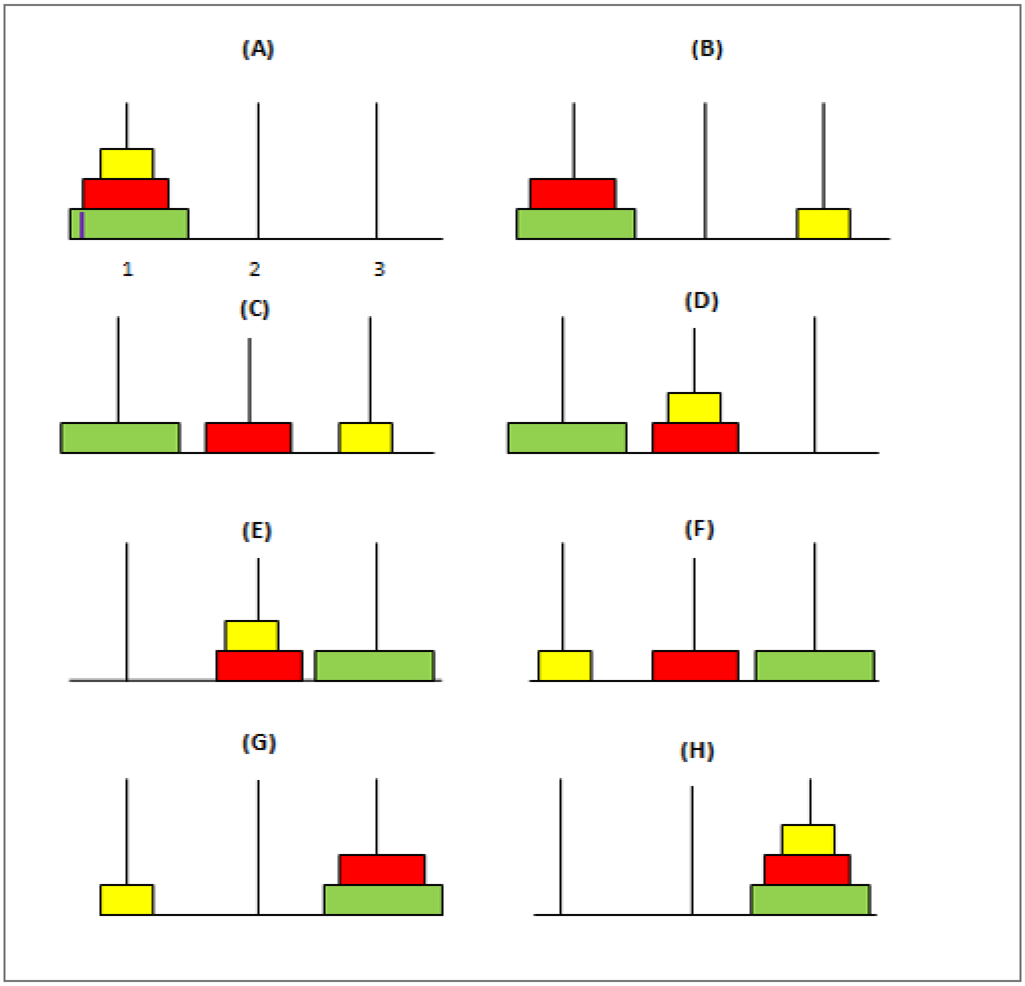
LAB-5

Q.1 This is a classic mathematical problem called “Tower of Hanoi” which is having 3 poles and an “n” number of disks with different sizes. The puzzle goes as follows:



In the beginning, the first pole will be having the disks arranged such that the biggest disc of them all is at the bottom and the smallest one at the top of the pole. The objective is to move these disks from the first pole to the third pole keeping the disks in the same position as that in the first. Following are a few conditions to keep in mind while shifting these disks:

- At a time, only one disk has to be moved.
- In the process, placing a larger disk over a smaller one is not allowed.
- The second (middle) pole can be used to mediate while transferring the discs from the first to the second pole.

Solve the above problem using recursion

Q.2 A set of “n” numbers is said to be in a Fibonacci sequence if number3=number1+number2, i.e. each number is a sum of its preceding two numbers. Hence the sequence always starts with the first two digits like 0 and 1. The third digit is a sum of 0 and 1 resulting in 1, the fourth number is the addition of 1 and 1 resulting in 2, and the sequence goes on.

Create a fibonacci sequence of length n (n is taken as user input) using recursion where first 2 elements of the sequence are 0 and 1.

Q. 3

```
public class Outer {
  void method1() {

    class InnerClass {

    }

    InnerClass ins = new InnerClass();
    ins.msg();
  }

  public static void main(String args[]) {
    Outer outer = new Outer();
    outer.method1();
  }
}
```

Define a function inside the inner class to print a message. Instantiate an inner class object from within method1 and call the newly defined function such that the output is:

```
This is an inner class
```