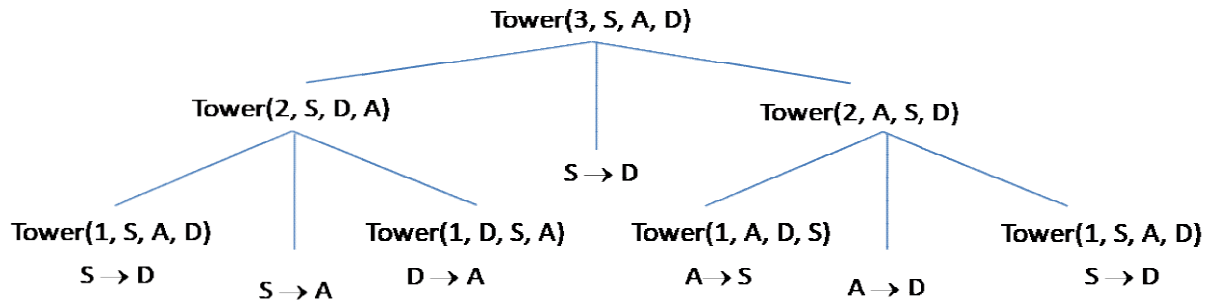


Recursion Tree of Tower of Hanoi Problem for n = 3.

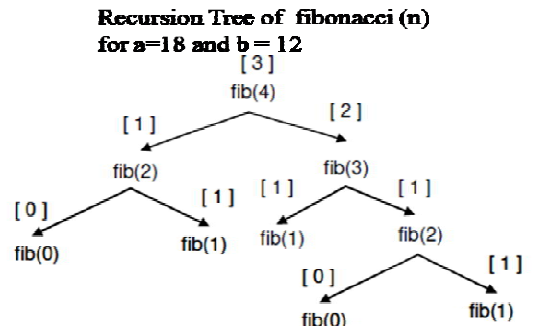
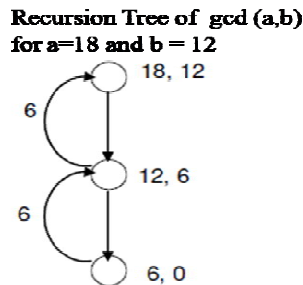
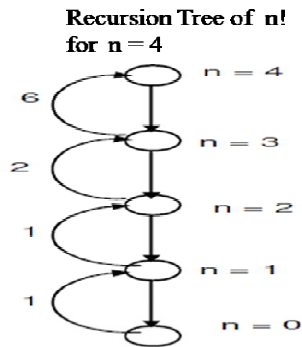
The following Recursion Tree shows solution to Tower of Hanoi Problem for n = 3. Carefully observe the function calls.



Recursive solution to Tower of Hanoi Problem for n = 3

The solution shows we require 7 moves to solve the problem recursively for n = 3. The moves are S → D, S → A, D → A, S → D, A → S, A → D, S → D.

=====



The return value of each function call is shown within third braces

Distinguish between Iteration and Recursion:

Sl	Iteration	Recursion
1	It is a process of executing statement(s) repeatedly, until some specific condition is satisfied.	It is a technique of defining anything in terms of itself.
2	Iteration involves four steps initialization, condition checking, execution of the statement(s) within loop body and updation of the control variable.	There must an exclusive “if” statement inside the recursive function, specifying stopping condition.
3	Any recursive problem can be solved iteratively.	Not all problems have recursive solution.
4	Larger code Size	Smaller code size
5	Fast in execution	Slow in execution due to function calling overhead.
6	<pre> int factorial_iteration(int n) { int prod = 1, i; for(i = 2; i <= n; i++) prod = prod * i; return(prod); } </pre>	<pre> int factorial_recursion(int n) { if(n == 0) return(1); return(n * factorial_recursion(n-1)); } </pre>