NON-RECURSIVE SOLUTION TO TOWER OF HANOI

We discussed problem of Tower of Hanoi earlier and written a recursive function to solve the problem, Recursive functions take lot of extra memory (New activation record for each call on the stack) (A detailed analysis of recursion is done in this post of mine).

Todays question is to write a Non-recursive function to solve problem of Tower Of Hanoi.

The function should not take more than O(n) time (n = number of Moves actually required to solve the problem) and O(1) extra space.

The signature of the function will be

```
/* The three char represents the characters representing three rods
 * and n is the number of discs (initially in s)
 */
 void towerOfHanoi(char s, char d, char e, unsigned int n)
```

Solution:

If there are n discs in a Tower Of Hanoi puzzle, then the total number of moves required to solve the puzzle will be $2^n - 1$.

The solution to this problem is required some moves to be repeated depending on whether n is even or odd and it is based on the below fact

At any given time, there is only one legal move between any two pegs.

Algorithm:

```
Repeat below steps till the total number of moves becomes 2^n - 1

If (n is even)

   Make_move (S, E)

   Make_move (S, D)

   Make_move (E, D)

Else

   Make_move (S, D)

   Make_move (S, E)

   Make_move (E, D)
```

The function Make_move is a simple function that will make the legal move between the two pegs (the only possible move)

Source: http://www.ritambhara.in/non-recursive-solution-to-tower-of-hanoi/