The National Institute of Engineering, Mysuru
(An Autonomous Institute under VTU)

ESTD : 1946

For the impartial fulfilment for the award of degree of

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING
by

Abhinav N B	**4NI21CS001**
Amogh P	**4NI21CS012**
Girish G M	**4NI21CS039**
Avi Kumar	**4NI21CS026**
Laksh Shetty	**4NI21CS053**
Tarun Raj	**4NI21CS033**
Ashrith V	**4NI21CS025**
Naveen Kumar	**4NI21CS060**
Akhil Kumar P	**4NI21CS010**

The National Institute of Engineering, Mysuru – 570008

Under the guidance of Dr.Anitha . R

Department of Computer Science Engineering
The National Institute of Engineering
Mysuru – 570008

INDEX

Tower Of Hanoi

Introduction

The Tower of Hanoi is a mathematical game or puzzle consisting of three rods and a number of disks of various diameters, which can slide onto any rod. The puzzle begins with the disks stacked on one rod in order of decreasing size, the smallest at the top, thus approximating a conical shape.

Origin

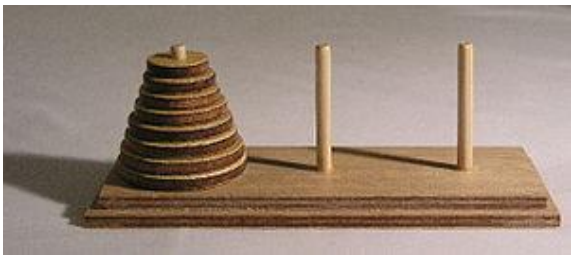
The puzzle was introduced to the West by the French mathematician Édouard Lucas in 1883. Numerous myths regarding the ancient and mystical nature of the puzzle popped up almost immediately, including one about an Indian temple in Kashi Vishwanath containing a large room with three time-worn posts in it, surrounded by 64 golden disks. Acting out the command of an ancient prophecy, Brahmin priests have been moving these disks in accordance with the immutable rules of Brahma since that time. The puzzle is therefore also known as the Tower of Brahma. According to the legend, when the last move of the puzzle is completed, the world will end.

If the legend were true, and if the priests were able to move disks at a rate of one per second, using the smallest number of moves, it would take them $2^{64} - 1$ seconds or roughly 585 billion years to finish, which is about 42 times the current age of the universe.

There are many variations on this legend. For instance, in some tellings, the temple is a monastery, and the priests are monks. The temple or monastery may be in various locales including Hanoi, and may be associated with any religion. In some versions, other elements are introduced, such as the fact that the tower was created at the beginning of the world, or that the priests or monks may make only one move per day.

Rules for solving Tower Of Hanoi

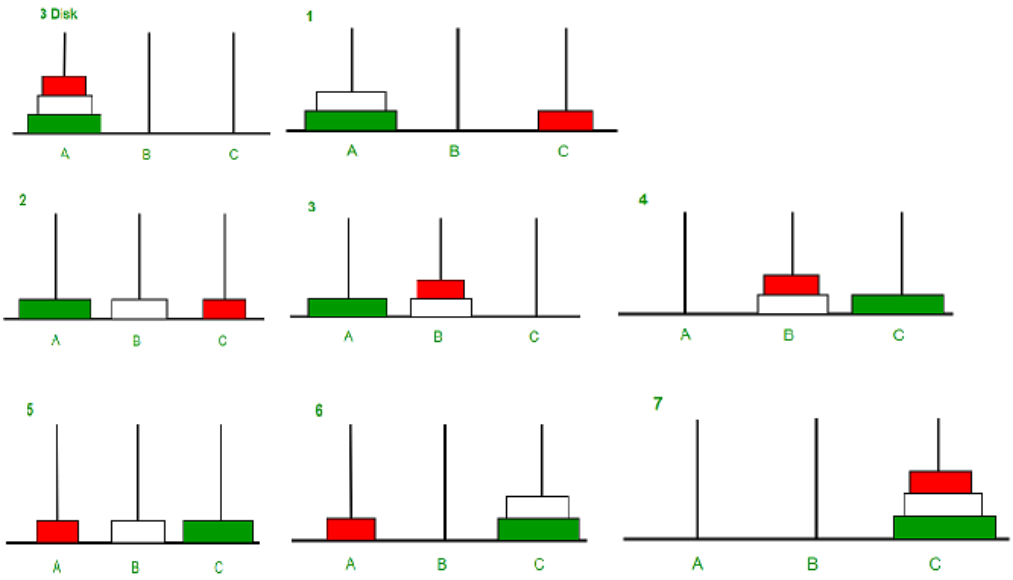
Only one disk can be moved at a time.
• Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
• No disk may be placed on top of a smaller disk.



Algorithm

The idea is to use the helper node to reach the destination using recursion. Below is the pattern for this problem:

- Shift ‘N-1’ disks from ‘A’ to ‘B’, using C.
- Shift last disk from ‘A’ to ‘C’.
- Shift ‘N-1’ disks from ‘B’ to ‘C’, using A.



Recurrence relation

For n=1, the number of moves are 1, i.e T(1)=1
For n=2, the number of moves are 3, i.e T(2)= 3 = 2T(1)+1
Similarly, for n=3 , T(3)= 7 = 2T(2) + 1
From the above illustration, it is clear that, it is a first order recurrence relation.
Therefore, in general,
T(n) = 2T(n-1)+1

Time Complexity

The time complexity of Tower of Hanoi can be analysed using a recurrence relation.
Let's assume that the time complexity of selection sort on an input array of size n is T(n).

Step 1: Size of problem is n
Step 2: Primitive operation is to move the disk from one peg to another peg
Step 3: Every call makes two recursive calls with a problem size of n – 1. And each call

corresponds to one primitive operation, so
recurrence for this problem can be set up as follows:

$T(n) = 2T(n – 1) + 1$ ...(1)

Let us solve this recurrence using forward and backward substitution:
Substitute n by n – 1 in Equation (1),

$T(n – 1) = 2T(n – 2) + 1$,
By putting this value back in Equation (1),

$T(n) = 2[2T(n – 2) + 1] + 1$
$= 2^2 T(n – 2) + 2 + 1$
$= 2^2 T(n – 2) + (2^2 – 1)$ ...(2)

Similarly, replace n by n – 2 in Equation (1),

$T(n – 2) = 2T(n – 3) + 1$,
From Equation (2),

$T(n) = 2^2 [2T(n – 3) + 1] + 2 + 1$
$= 2^3 T(n – 3) + 2^2 + 2 + 1$
$= 2^3 T(n – 3) + (2^3 – 1)$
$T(n) = T(n-1) + n-1$

In general,
$T(n) = 2^k T(n – k) + (2^k – 1)$
By putting $k = n – 1$,

$T(n) = 2^{n–1} [T(1)] + (2^{n – 1}– 1)$
T(1) indicates problem of size 1. To shift 1 disk from source to destination peg takes only one move, so T(1) =1.

$T(n) = 2^{n–1} + (2^{n – 1}– 1)$
$= 2^n – 1$

Thus, $T(n) = O(2^n)$

There is no meaningful best- or worst-case time complexity, Time complexity increases, as the value of n increases.

Code:

```c
/* Tower of Hanoi: Recursive implementation */
#include <stdio.h>

// C recursive function to solve tower of hanoi puzzle
void towerOfHanoi(int n, char from_rod, char to_rod, char aux_rod)
{
    if (n == 1)
    {
        printf("\nMove disk 1 from rod %c to rod %c", from_rod, to_rod);
        return;
    }
    towerOfHanoi(n-1, from_rod, aux_rod, to_rod);
    printf("\nMove disk %d from rod %c to rod %c", n, from_rod, to_rod);
    towerOfHanoi(n-1, aux_rod, to_rod, from_rod);
}

int main()
{
    int n;// Number of disks
    printf("Enter number of discs:");
    scanf("%d", &n);
    printf("Moves for the solution are:");
    towerOfHanoi(n, 'A', 'C', 'B'); // A, B and C are names of rods
    return 0;
}
```

Output:

```
Enter number of discs:4
Moves for the solution are:
Move disk 1 from rod A to rod B
Move disk 2 from rod A to rod C
Move disk 1 from rod B to rod C
Move disk 3 from rod A to rod B
Move disk 1 from rod C to rod A
Move disk 2 from rod C to rod B
Move disk 1 from rod A to rod B
Move disk 4 from rod A to rod C
Move disk 1 from rod B to rod C
Move disk 2 from rod B to rod A
Move disk 1 from rod C to rod A
Move disk 3 from rod B to rod C
Move disk 1 from rod A to rod B
Move disk 2 from rod A to rod C
Move disk 1 from rod B to rod C
Process returned 0 (0x0)   execution time : 6.983 s
Press any key to continue.
```

Application:

Real World Applications
While the Tower of Hanoi's past and present mainly involve recreational math, its future involves major real world applications. The Tower of Hanoi game can be used to assess the extent of various brain injuries and it also acts as an aid to rebuild neural pathways in the brain and to forge new connections in the prefrontal lobe. Attempting to solve the Tower of Hanoi exercises parts of the brain that help to manage time, present a business plan or make complex arguments. Even without actually solving the puzzle, anyone who attempts to solve the Tower of Hanoi can benefit (entertainment).

Not only is the Tower of Hanoi beneficial in physical and mental terms, but also in terms of certain jobs. The Tower of Hanoi is commonly used by psychologists to research and examine problem solving skills (funtrivia). Problem solving skills can be acquired by calculating moves and strategies while at the same time predicting possible outcomes (entertainment). The recursive rule of the Tower of Hanoi is studied and applied in computer programming and algorithms which helps to reduce the amount of time it takes to create a program (funtrivia).

Pile Problems Applications in Logistics Management Mathematical Modeling
Puzzles are connected with graphs, it means can be modeled by using graphs, vertices of the graph represent configuration and graph edges define possible moves of the puzzle (Cull, Merril and Van, 2013). According (Hempel & Schmiedel, 2006) to describe pile problems more precisely is highly recommended using graph theory. For these authors:

Conclusion:

Tower of Hanoi is a simple mathematical puzzle, which works at time complexity $O(2^n)$, the time taken by the puzzle mainly depends on the number of disks used, that is the value of n, so there isn't no worst or average case in it, we explored the time complexity and recurrence relation of Tower of Hanoi and discussed some of its application. We also provided a simple implementation of tower of hanoi in C and an example of how it works.