**EXPERIMENT NO. 4**

| **Objective(s):**  The objective is to move a stack of disks from one rod to another. |
| --- |
| **Outcome:**  The outcome is to solve the puzzle using the minimum number of moves. The minimum number of moves required to solve a Towers of Hanoi puzzle with n disks is 2^n - 1. |
| **Problem Statement:**  Implement Towers of Hanoi using Stack. |
| **Background Study:**   1. **Origin and History:**  * The Towers of Hanoi puzzle was invented by the French mathematician Édouard Lucas in   1883. Lucas named the puzzle after the Tower of Hanoi, a shrine in Vietnam where the legend of the puzzle originates.   * It was first introduced to the public in a recreational mathematics column in the French newspaper *L'Echo de Paris.*  1. **Problem Description:**  * The Towers of Hanoi puzzle consists of three rods and a number of disks of different sizes that can slide onto any rod. * The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, making a conical shape.  1. **Objective:**  * The objective is to move the entire stack to another rod, obeying the following simple rules:   + 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 or on an empty rod.   + No larger disk may be placed on top of a smaller disk.  1. **Minimum Moves**:  * The minimum number of moves required to solve the puzzle with nnn disks is 2^n - 1. This exponential growth makes the puzzle interesting as n increases.  1. **Mathematical and Computational Aspects:**  * **Recursive Solution**: The problem can be solved recursively. The recursive solution is elegant and can be described as:   + Move n−1 disks from the source rod to the auxiliary rod.   + Move the nth disk (the largest one) directly to the destination rod.   + Move n−1n-1n−1 disks from the auxiliary rod to the destination rod. * This recursive approach demonstrates the power and elegance of recursion in solving problems that can be broken down into smaller, similar problems |

| **Algorithm (Student Work Area):**  **Function t\_o\_h:**   * This function is defined to solve the Towers of Hanoi problem recursively. * Parameters:   + n: Number of disks to be moved.   + s, a, d: Characters representing the three rods ('A', 'B', 'C' in this case).   + s: Source rod.   + a: Auxiliary rod.   + d: Destination rod.   **Base** **Case (n == 1):**   * If there is only one disk (n == 1):   + Move the disk from the source rod (s) to the destination rod (d).   + Print: Move disk 1 from rod <source> to rod <destination>.   + Return from the function.   **Recursive Case (n > 1):**   * Move n-1 disks from the source rod (s) to the auxiliary rod (a), using the destination rod (d) as auxiliary:   + Call t\_o\_h(n - 1, s, d, a);. * Move the nth disk (the largest one) from the source rod (s) to the destination rod (d):   + Print: Move disk <n> from rod <source> to rod <destination>. * Move n-1 disks from the auxiliary rod (a) to the destination rod (d), using the source rod (s) as auxiliary:   + Call t\_o\_h(n - 1, a, s, d); |
| --- |
|  |

| **Code:** |
| --- |
| **OUTPUT :** |