Name: Hana Saeed Mansour

ID: 202301462

Task two analysis

The tower of Hanoi is a classic mathematical puzzle that involves three pegs and several disks of different sizes. The goal is to move all the disks from one peg to another peg. Usually from A to C and B is used to help. The rules are that only one disk can be moved at a time, and no larger disk may ever be placed on top of a smaller one.

The restricted tower of Hanoi is a harder version of the normal tower of Hanoi puzzle, in the normal version, there is one extra rule, every move must either go to peg B or from peg B. this means you can’t move a disk directly from A to C to A. because of this, the puzzle takes more moves to finish.

The algorithm used in the program is recursive, meaning that it solves the problem by breaking it into smaller versions of itself. To move n disks from A to C:

1- Move n-1 disks from A to C. recursively, using the same rule.
2- Move the largest disk from A to B
3- Move the n-1 from C back to A
4- Move the largest disk from B to C
5- Move the n-1 disks from A to C again

This pattern keeps repeating until only one disk is left, which can be moved from A to B and then from B to C. the total number of moves follows the formula r(n) = 3r(n-1) + 2 and when simplified, it becomes 3^n – 1. This is higher than the normal tower of Hanoi’s 2^n – 1 moves because of the restriction on how disks can move.

Other ways to solve this could include the normal tower of Hanoi algorithm or an iterative version that uses loops instead of recursion. The iterative version would work the same way but would manually track the steps instead of using functions calls. The recursive version is easier to understand and matches the problems logic more naturally.

In the code. The Main() function is the starting point. It asks the user for the number of disks, sets up three pegs, A, B, and C, then fills peg A with disks in order from largest to smallest, then calls the recursive function restrictedHanoi() to start solving the puzzle and shows how many moves it took in total.

restrictedHanoi() is the main recursive function, it follows the five steps above to move the disks according to the restricted rules. It calls itself to handle smaller problems until only one disk is left.

moveDisk() is a helper function that moves one disk from one peg to another. It also checks that the move is valid, meaning it won't allow a bigger disk on a smaller one and counts each move. It can also print each move if visualization is turned on.

printPegs() simply shows the current state of the three pegs after each move so the user can see how the disks move throughout the puzzle.

The time complexity of the algorithm is O(3^n) because each level of recursion makes three smaller recursive calls. The space complexity is O(n) because of the recursion stack. In an iterative version, the time complexity would still me O(3^n), but it would not use the function call stack, instead using loops and manual tracking.

Overall, the recursive restricted Tower of Hanoi is a clear example of how adding one small rule can make a simple puzzle more complex. The code shows how recursion can be used to handle problems that repeat a similar pattern at smaller scales.

References :

GeeksforGeeks. (2025, September 3). *Program for Tower of Hanoi algorithm*. GeeksforGeeks. https://www.geeksforgeeks.org/dsa/c-program-for-tower-of-hanoi/