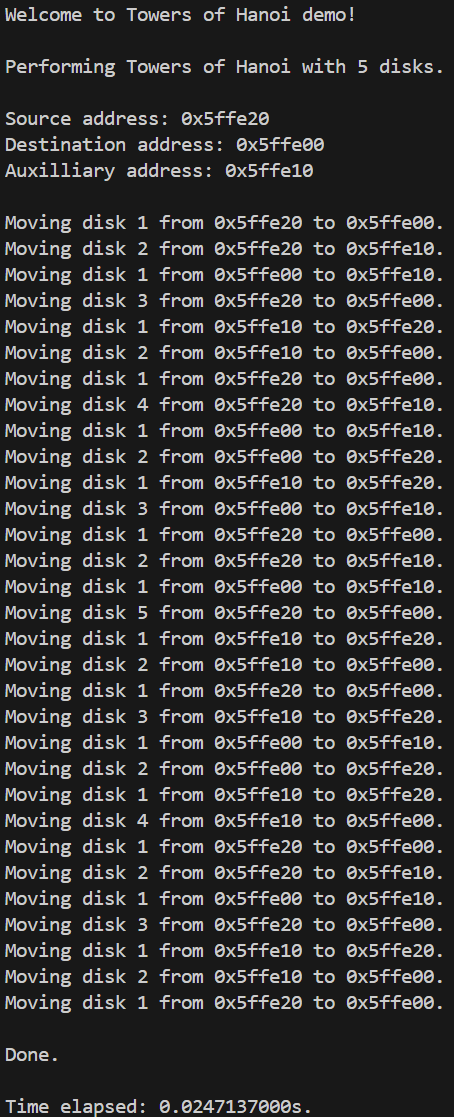
What do you infer from this experiment? Submit screenshots of the results, code files, and a write-up describing your findings in a zip file.



**Figure 1.** Console output from recursive Towers of Hanoi algorithm implementation in C++ on an input size of 5. Note the time duration is exaggerated by the inclusion of the print statements. For 5 disks, 31 moves are required (2n - 1) in the optimal implementation, as shown.

**Table 1.** Table of time durations from executing Towers of Hanoi algorithm on various input sizes.



The estimated execution time for an input size of 125 disks can be determined as follows:

This is times the estimated age of the universe. In other words, the exponential time complexity of the Towers of Hanoi algorithm causes execution time to blow up remarkably fast. Based on the apparent simplicity of the algorithm, I would not have expected this.

The implementation uses array-based stacks to physically store the hypothetical disks. A source stack is generated to contain integer elements in increasing order from top to bottom, representing an increasing order of disk sizes. Auxiliary and destination stacks are also inputs to the algorithm, allowing verification of the implementation’s correctness.