Analysis of Algorithms Laboratory

Towers of Hanoi: Complexity Analysis

Aispur P. Santiago
School of Mathematical and Computational Sciences
Yachay Tech University
San Miguel de Urcuquí - Imbabura, Ecuador
March 2, 2026

1 Introduction

This report presents the implementation and analysis of the Towers of Hanoi problem through two approaches: recursive and iterative algorithms. The objective is to verify the consistency between mathematical and empirical complexity analysis.

2 Mathematical Analysis

The Towers of Hanoi problem requires moving n disks from one rod to another following specific rules. The number of moves follows a recurrence relation:

$$M(n) = M(n-1) + 1 + M(n-1) = 2 \cdot M(n-1) + 1 \tag{1}$$

With base case $M(1) = 1$, the closed form is:

$$M(n) = 2^n - 1 \tag{2}$$

The time complexity is $O(2^n)$, indicating exponential growth.

3 Implementation

Two algorithms were implemented in C++:

Recursive approach: Follows the classic divide-and-conquer strategy. To move n disks from source to destination:

- Move $n-1$ disks from source to auxiliary
- Move the largest disk from source to destination
- Move $n-1$ disks from auxiliary to destination

Iterative approach: Uses a cyclic pattern based on the parity of n. For odd n, the sequence is A→C, A→B, B→C. For even n, the sequence is A→B, A→C, B→C.

Both implementations utilize a stack-based data structure to represent the rods.

4 Empirical Analysis

Execution time was measured for input sizes $n = 1$ to $n = 25$ with the C++ `<chrono>` library. Tests were run with console output disabled to measure only the algorithmic operations. Table 1 shows selected results.

Table 1: Empirical results for selected input sizes

n	Theoretical Moves	Iterative (ms)	Recursive (ms)
5	31	0.002	0.001
10	1,023	0.043	0.011
15	32,767	0.883	0.509
20	1,048,575	13.135	6.190
25	33,554,431	400.216	228.290

Figure 1 shows the empirical analysis graphs comparing both algorithms.

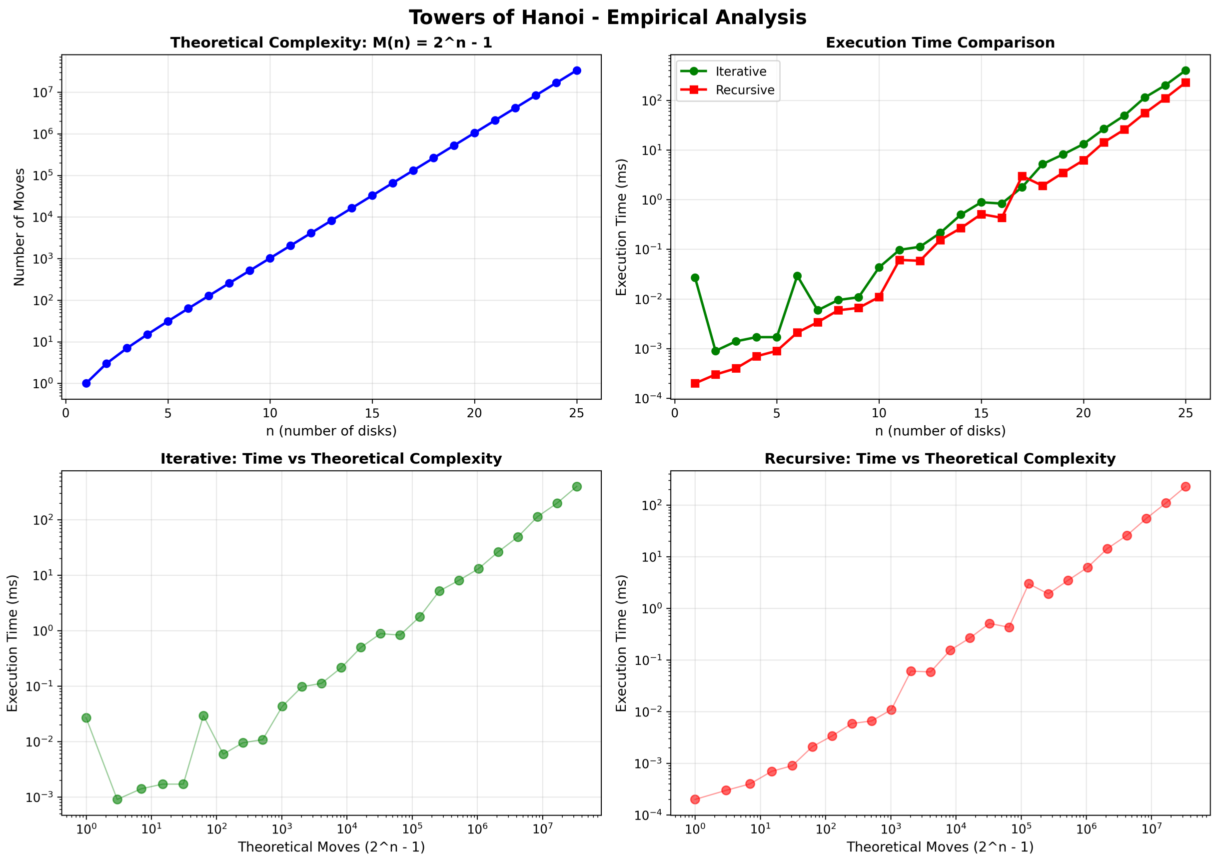

Figure 1: Empirical analysis of Towers of Hanoi algorithms

5 Results and Discussion

The empirical data confirms the mathematical prediction of $O(2^n)$ complexity:

- **Exponential growth:** Execution time doubles approximately with each increment of n
- **Consistency:** The measured time correlates with $M(n) = 2^n - 1$

- **Performance comparison:** The recursive implementation runs faster for larger n due to better cache locality, despite function call overhead

The logarithmic plots in Figure 1 show linear trends, confirming the exponential nature of both algorithms. Both implementations perform the same number of operations but differ in execution time due to implementation details.

5.1 Verification

For $n = 25$:

- Theoretical moves: $2^{25} - 1 = 33,554,431$
- Iterative time: 400.22 ms
- Recursive time: 228.29 ms

The growth rate between consecutive values confirms $M(n) = 2 \cdot M(n-1) + 1$. For example, $M(25) = 33,554,431 \approx 2 \cdot M(24) + 1 = 2 \cdot 16,777,215 + 1$.

6 Conclusion

The mathematical and empirical analyses agree. Both the recursive and iterative implementations of the Towers of Hanoi problem exhibit $O(2^n)$ time complexity. The empirical measurements match the theoretical prediction $M(n) = 2^n - 1$ across all tested input sizes from $n = 1$ to $n = 25$, with a maximum of 33,554,431 moves computed in under 500 milliseconds.

Repository

Full source code available at: Aispur Santiago -Analysis of Algorithms, GitHub Repository