**Time Complexity:**

**Recursive Approach:** O(n) time complexity since the function performs a constant amount of work in each of the n recursive calls.

**Space Complexity:**

**Recursive Approach:** O(n) due to stack usage.

**Iterative Approach:** O(1), which is more memory efficient.

**Optimizations to Avoid Excessive Computation:**

Convert the recursion to an iterative solution to reduce stack overhead.

Although tail recursion may theoretically help, it is not supported in many languages like Java.

Use memoization in cases with overlapping subproblems; however, it doesn't apply here.