

## **Explanation of Differences (Advantages, Disadvantages) between the 3 Implementations:**

### **Iterative (Loop-based) Implementation:**

**Advantages:** This implementation is easy to understand and does not use additional function calls, which can save memory space.

**Disadvantages:** It may require more code and variables to keep track of intermediate results, especially for larger values of  $n$ .

### **Recursive Implementation:**

**Advantages:** The recursive implementation is straightforward, closely following the mathematical definition of the recurrence relation.

**Disadvantages:** It can be inefficient and lead to redundant calculations due to multiple function calls with the same inputs. For large values of  $n$ , it can result in a significant performance hit and may cause a stack overflow if the recursion depth becomes too large.

### **Memoization Implementation:**

**Advantages:** Memoization improves the performance of the recursive approach by storing previously computed results in an array (memo) and reusing them when needed, reducing redundant calculations.

**Disadvantages:** Although it performs better than the pure recursive approach, it requires additional memory to store the memoization array, which can be a concern for very large values of  $n$ .

Overall, the iterative implementation is preferable when memory efficiency is crucial and the value of  $n$  is relatively small. The recursive implementation is the most straightforward to understand but may need to be more efficient for large values of  $n$ . The memoization approach strikes a balance between efficiency and readability by avoiding redundant calculations at the expense of some additional memory usage. In practice, for larger values of  $n$ , the memoization implementation is the most efficient option.

**Submitted to:** HR Bitnine Global

**Submitted by:** Uzair Sarfraz