CALIFORNIA STATE UNIVERSITY, FRESNO

DEPARTMENT OF COMPUTER SCIENCE

Class:	**Algorithms & Data Structures**	Semester:	**Fall 2023**

Points		Document author:	**Pratham Aggarwal**
		Author's email:	**pratham_aggarwal@mail.fresnostate.edu email**
		Laboratory number:	**2**

1. Statement of Objectives

The objective of the lab is to find the efficiency and the time required to run the algorithm having same result but by different methods. This experiment's objectives include putting both algorithms into practice, evaluating how quickly they run, and comparing the outcomes. The significance of this lab lies in understanding the importance of algorithmic efficiency and its impact on computation time.

2. Experimental Procedure

Inefficient Algorithm (Recursive): Recursively, or until it reaches the base case, this algorithm calculates Fibonacci numbers by making smaller calls to itself over and over. When n is 0 or 1, the base case, it returns n. If not, it computes Fibonacci(n-1) and Fibonacci(n-2) and adds them.

Efficient Algorithm (Iterative): The Fibonacci numbers are calculated using an iterative process by this algorithm. The first two variables (a and b) are initialized to 0 and 1, respectively, at the beginning and a variable

‘result’ is declared to store the current added value. The Fibonacci numbers from 2 to n are then calculated iteratively using a loop. In order to calculate the subsequent Fibonacci number until it reaches the required nth Fibonacci number, it changes the values of a, b and result for each iteration.

3. Analysis

As we can see, the 40th Fibonacci number, 102334155, is calculated precisely by both algorithms. There are different execution times, though. The time required by the ineffective recursive method is roughly 1.02805 seconds, whereas the time required by the effective iterative algorithm is almost 0 seconds. This indicates that in this particular example, the efficient iterative technique is faster than the inefficient recursive algorithm. While in some instances, such as for the 10th number, both results are zero, we can conclude that as the number grows, the execution time for inefficient algorithms increases while the execution time for efficient algorithms stays zero, for a big number, such as 55^{th} as well, but may change for more larger numbers.

Screenshots of output in the end of the report

4. Encountered Problems

As such I didn’t encountered problem in understanding the algorithm, though I had to look up for the Inefficient Recursive Algorithm as I was making one mistake in the recursive function part.

5. Conclusions

In conclusion, the purpose of this lab was to examine the effectiveness of two algorithms for computing the Fibonacci sequence: an ineffective recursive algorithm and an effective iterative algorithm. The study of the results demonstrates that the efficient iterative approach for computing Fibonacci numbers executes substantially more quickly than the inefficient recursive algorithm. This emphasizes how crucial algorithmic effectiveness is

to the optimization of larger algorithms. So, for larger algorithms we can say that efficient algorithms are preferred since they can drastically reduce execution time for bigger algorithms.

6. References

N/A