# California State University, Fresno

# DEPARTMENT OF COMPUTER SCIENCE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class: | **Algorithms & Data Structures** | | | Semester: | **Fall 2021** |
|  | | | | | |
| Points |  | Document author: | **Saishnu Ramesh Kumar** | | |
|  | Author’s email: | **saishnu\_rk@mail.fresnostate.edu** | | |
| Laboratory number: | **Section 1, 11am to 12:50pm** | | |
|  | | | | | |

**1. Statement of Objectives**

This lab exercise deals with the Fibonacci sequence of numbers and how to implement them in the recursive form as well as either using arrays or a temporary variable. The experiment was to allow the user to input any integer-based number into the program and it would output its respective number from the sequence. This report will also include a table to compare the two algorithms that were used to find the 120th number in the sequence as the time taken to get the result may differ depending on the algorithm.

**2. Experimental Procedure**

For the first part, it was really simple to create a recursive function as a recursive function is meant to call itself within the program. A fibRec function was created and it included the code to either return n if no other inputted number was present otherwise, it would return the value of the inputted number’s Fibonacci sequence for example if the user were to input the number 5, the number in the sequence would be 5. As for the second problem, this required using a temporary variable to store the number and do some switching among the numbers to get the result. Like in a basic switching exercise, you would need to include a temp variable so that it is easy to switch the numbers around. These both were approached similarly as the result was to get the user’s inputted number from the sequence. The best algorithm among the two would be the second one where the temporary variable was used because the execution time is much faster than the traditional recursion function. So, it can be said that the 2 variable program is far more efficient than the recursive function in respect to its execution time.

**3. Analysis**

These are the results from the experiment:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Function Name | Implementation Type | Time Complexity | Input and Output 1 | Time 1 (seconds) | Input and Output 2 | Time 2 (seconds) | Input and Output 3 | Time 3 (seconds) |
| fibRec | Recursion | O(2n) | 25 = 75025 | 0.0006882 | 50 = 12586269025 | 94.476 | 120 = no output due to overflow |  |
| fibVar | Temporary Variable | O(n) | 25 = 75025 | 0.0000003 | 50 = 12586269025 | 0.0000003 | 120 = 4376692037216111008 | 0.0000006 |
|  |  | Average Time = |  | 0.00034425 |  | 47.2380002 |  |  |

\*Note: The value for the 120th term in the fibVar section is incorrect as there is overflow in the program.

**4. Encountered Problems**

During the experiment, the issue that was faced was trying to get the 120th term but it resulted in either the program continuously run without output or just cause an overflow. I attempted to use a different datatype that can handle a much larger group of numbers, in this case, the long long datatype was used but it was still not possible. After doing some research, once the user inputs the 94th term, there will be an overflow because the output data is higher than what the datatype can hold. You may need to use a third-party library to get any results higher than the 94th term. Any number below the 94th term was obtainable as seen in the table above.

**5. Conclusions**

To conclude, I have learned more about how to implement a recursive function as well as using a temporary variable to switch between each number to get the result of a Fibonacci number. Moreover, in this experiment, I have realized the difference between the maximum values of different datatypes because as mentioned above, all datatypes have a set of ranges for their minimum and maximum lengths which is why attempting to get the 120th term from this program was not possible unless you were to use an external library not offered in C++. Nonetheless, learning how to find the numbers within the Fibonacci sequence by using different algorithms was interesting.

**6. References**

The slides provided by the TA during the lab session.

<https://www.geeksforgeeks.org/program-for-nth-fibonacci-number/>

<https://spiderlabweb.com/cpp-program-print-fibonacci-sequence/>