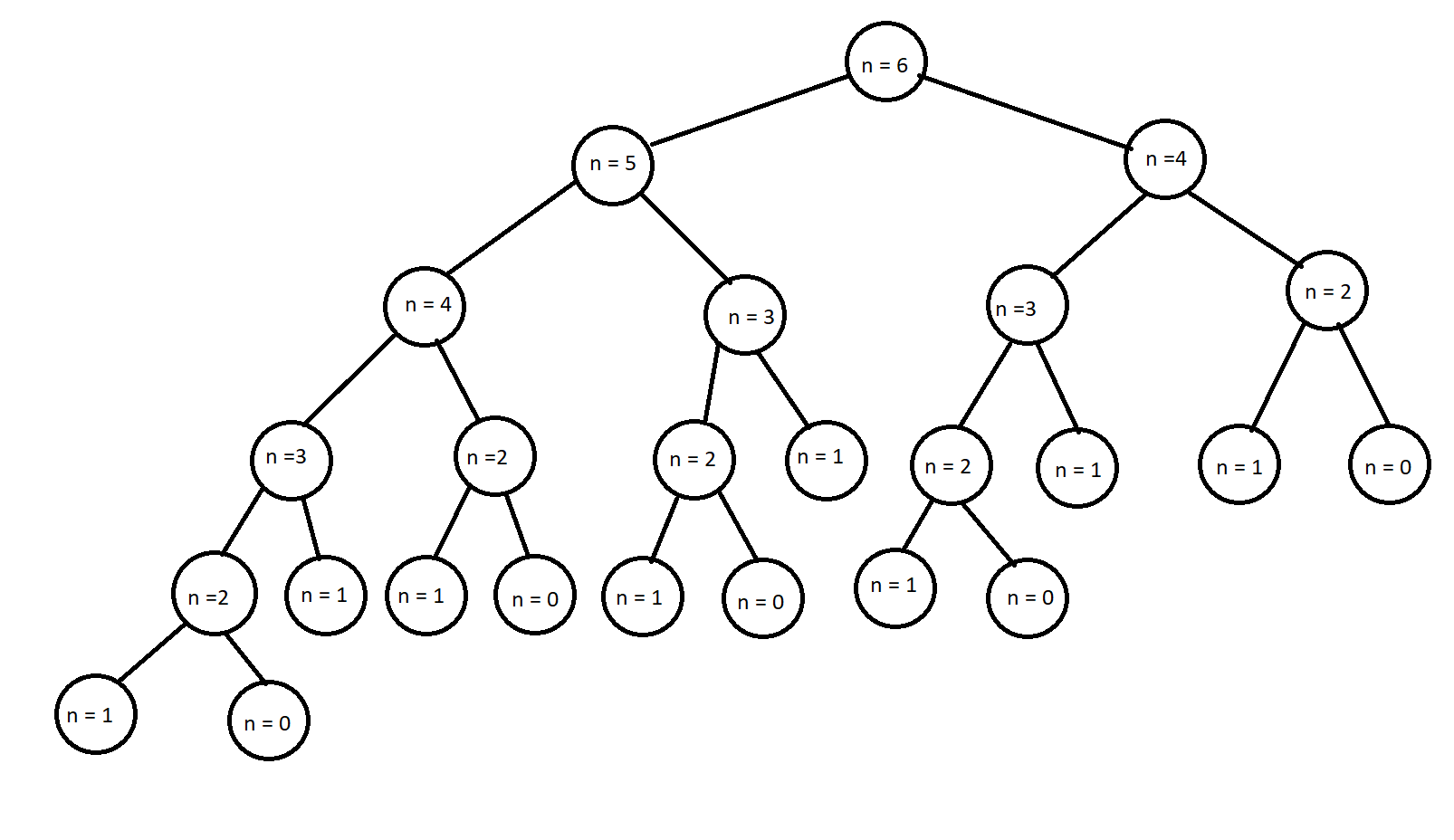
Antonio Santos

30046544

Questions:

1. 1. Yes, the algorithm performs redundant calculations. As shown in the recursive tree in 1(b), there are many instances where the function returns the same value due to the value of n being the same in many calls.
   2. 
      1. F­0 was called 5 times
      2. F1 was called 8 times
      3. F2 was called 5 times
      4. F3 was called 3 times
      5. F4 was called 2 times
      6. F5 was called 1 time

1. The big-O running time for algorithm 2 is O(n) which follows the trend of a linear graph. It is linear because since the function is using a loop, the time required to complete the function grows directly with the size of the problem.
2. The big-O running time for algorithm 3 is O(log(n)) which follows the trend of a logarithmic graph. As the value of n increases, the time it takes to execute the function increases slowly.
3. For the first algorithm, it would be optimal to use it with small values of n since as it gets bigger, the time required to complete the function gets exponentially greater. The second algorithm would be used for most values since the time required grows exponentially. As for the third algorithm, it would be recommended when the value of n is very great as it does not take as long as the other two algorithms to perform the same task.

Figure 1: plot of the time taken for Algorithm 1 to produce the nth value of the Fibonacci series

Figure 2: plot of the time taken for Algorithm 2 to produce the nth value of the Fibonacci series

Figure 3: plot of the time taken for Algorithm 3 to produce the nth value of the Fibonacci series