Problem 0

1.list out the the function call stack

fib(5) -> fib(4) -> fib(3) -> fib(2) -> fib(1) -> fib(0)

fib(1) -> fib(0)

fib(2) -> fib(1) -> fib(0)

-> fib(0)

fib(3) -> fib(2) -> fib(1) -> fib(0)

fib(1) -> fib(0)

-> fib(1)

2. Prove the time complexity of the algorithms.

For the equation the time complexity will be

T(n)=T(n-1)+T(n-2)+1

For each value of n, the function makes two recursive calls with parameters n-1 and n-2. This branching continues until it reaches n = 0 or n = 1, at which the recursion stops. The number of function calls grows exponentially with the value of n because each function call results in two more functions $O(2^n)$

3. Comment on way's you could improve your implementation

We can improve this by storing the values of fib(4), fib(3)……. Where these values can be reused instead of recalculation and calling all the below functions