Question: https://leetcode.com/problems/fibonacci-number/submissions/

We all are taught at some point of time that what is Fibonacci series,

f(n)=f(n-1)+f(n-2).

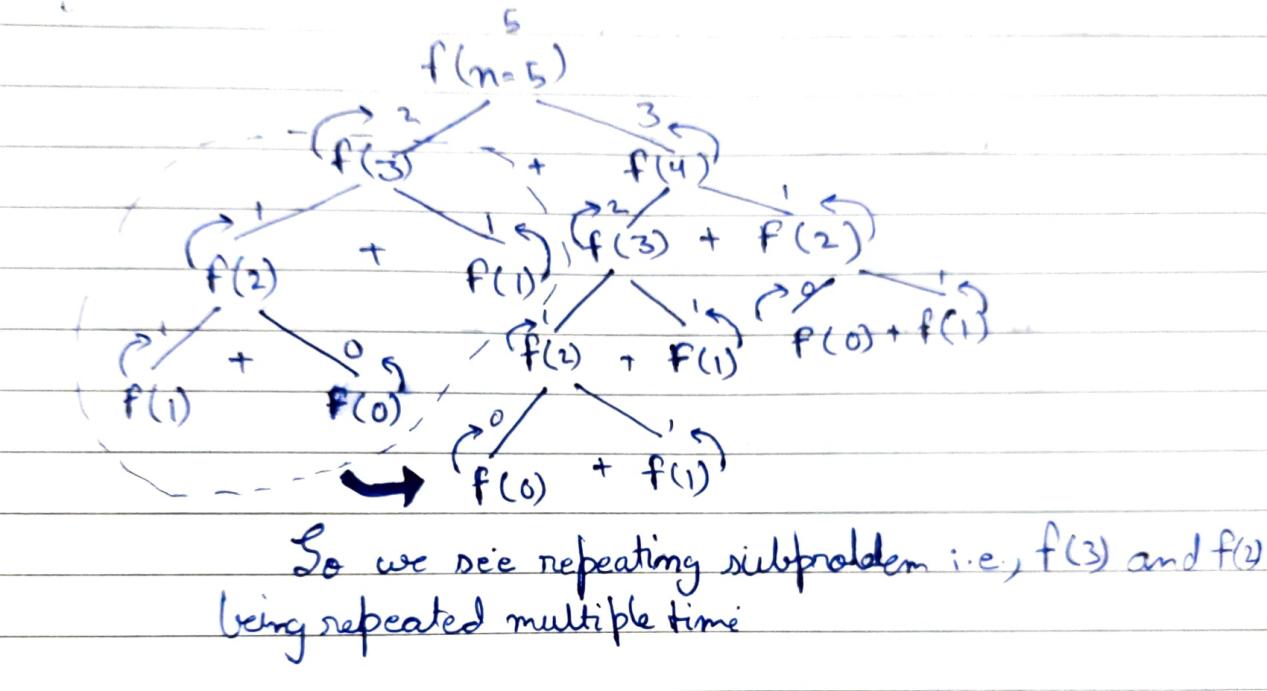
Generally we use an recursive approach to solve it that is we describe an function named fibo(int n) of return type integer.

Where we return fibo(n-1)+fibo(n-2), and the base cases to break the recursion are:

1. if n=0 return 0
2. If n=1 return 1

With that being said lets take a look what are the down side to this approach

If we draw a tree for finding 5th Fibonacci number, we will see that there are repeating sub problems.



So in order to reduce this problem what we can do is use a HashMap and memorize the value of a sub problem, so that in case if it occurs again we can directly get it from the Map itself.

Solution:  
class Solution {

HashMap<Integer, Integer> mem = new HashMap<>();

public int fib(int n) {

if(n==0){

return 0;

}else if(n==1){

return 1;

}else if(mem.containsKey(n)){

return mem.get(n);

}else{

mem.put(n,fib(n-1)+fib(n-2));

return mem.get(n);

}

}

}

Github Link :<https://lnkd.in/ecwtJeaz>