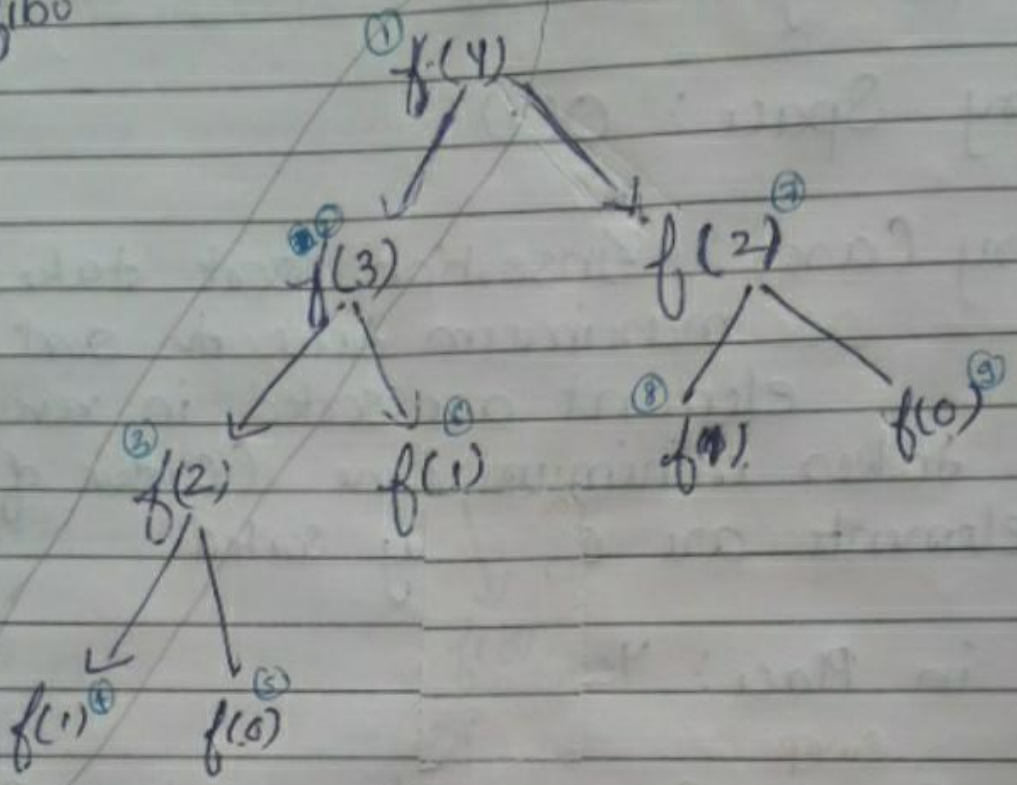
Pre requistive : Recursion of fibonacchi number

* Recursive algorithms

Masters Theorem

① fibo



O(n)

Q what is the space complexity ?

Ans:-

1) Sop Space complexity is not going to be constant in Recursive program because

a) In Recursion we know that functions calls are stored in stack.

b) So, those function call actually takes some memory in stack

2) Therefore Recursive program do not have constant space in complexity

3) So, the space complexity is actually the height of the tree

"At any particular point of time NO two funct^n call at perform at the same level of recursion will be in the stack at the same time.

Points:- Space complexity of recursive programs:-

1) When we talked about the flow of recursive program do you think that at any particular level

eg:- ③ ↙ ↘ ⑥ ⑧ ↙ ↘ ⑨
f(2) & f(1) f(1) f(0)

is there going to be a possibility for more than one function calls in this level to be in the stack at the same time?

2) Can we say that f(2) & f(0) are present in stack at the same time? NO.
Because this f(0) not even executed until f(2) finishes.

3) Please refer the fig:-
So, this 7, 8, 9 will for eg, will only execute when the ans of 3 is given.

Trick:-
Only calls that are interlinked with each other will be in the stack & at the same time.
because the previous one will be waiting for the next one to execute, next one will be wait for the next one ..... and these should be interlinked together at the same time.

from the fig:-

① for eg. when the function call no. 7 will be executing fun^{n} call no 2, 3, 4, 5, 6 would have already been finished.

② So, we suppose we take function call no 7. then 2, 3, 4, 5, 6 would have been already executed. Why?
Because those are not linked with function call no. 7

③ At one particular level there won't be any more than one calls that are in the stack at the same time.

④ So, what is the maximum space that it'll be taking?
Ans: We know that the longest st chain starts from the root till the leaf.
So, the longest chain will be the answer.

∴ Space complexity = height of the tree or path

∴ On

⑤ So, what is the auxiliary space complexity required when we're calculating Nth fibonacci number?
Ans O(n)
Because these all will be in the stack at the same time. so, the maximum amt of space req^{d} at the time.