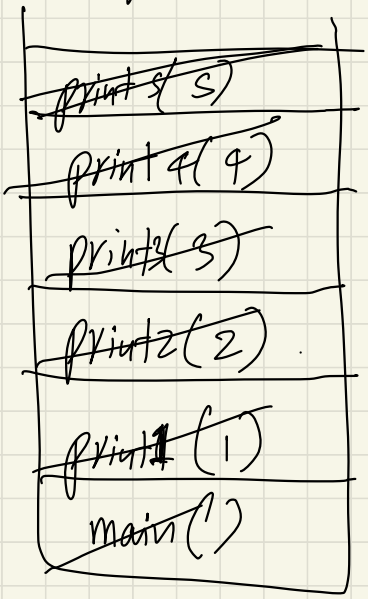
How function calls work in languages:



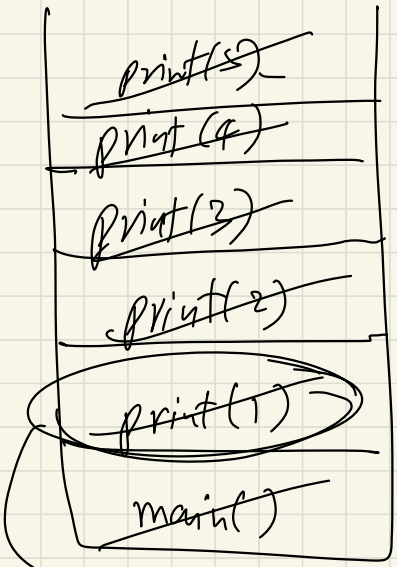
Output
1
2
3
4
5

Recursion: function calling itself.

VVI

- While the func is not finished executing, it will remain in stack.
- When a function finishes executing, it is removed from stack, and the flow of program is restored to where the function was called

Recursive function for the same:



1
2
3
4
5
6

→ every call of function, will take some memory

★ Base condition in Recursion:

Condition where our recursion will stop making new calls.

No base condition → function calls will keep happening, stack will be filled again and again

→ Memory of computer will exceed the limit → Stack overflow error

Why Recursion ?

Ans : ✯ It helps us in solving bigger / complex problems in a simple way.

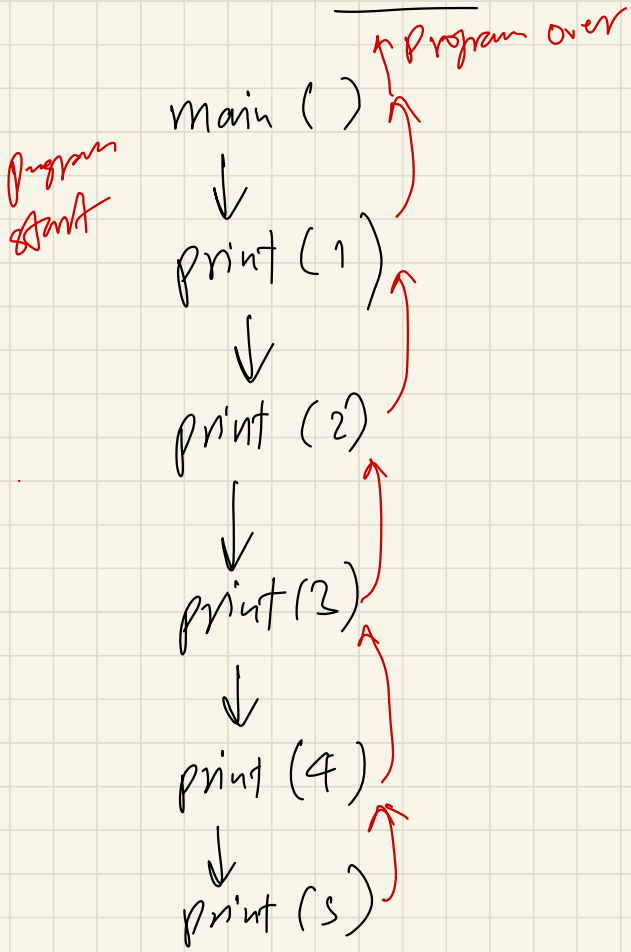
✯ You can convert recursion sol^{n} into iteration & vice versa.

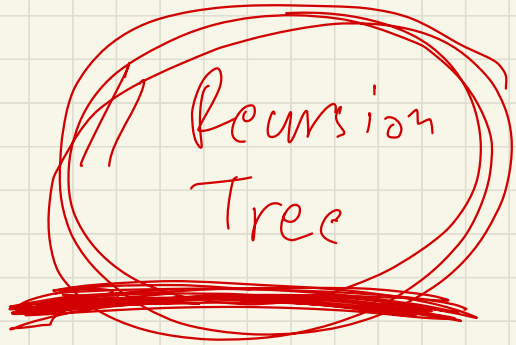
✯ Space complexity is not constant because of recursive calls.

✯ It helps us in breaking down bigger problems into smaller problems.

Visualising Recursion

VVVVI





Q: Find n^{th} fibonacci number.

0^{th} 1^{st} 2^{nd} 3^{rd} 4^{th} 5^{th} 6^{th} 7^{th}

0, 1, 1, 2, 3, 5, 8, 13 . . .

$$Fibo(N) = Fibo(N-1) + Fibo(N-2)$$

This is known as recurrence relation

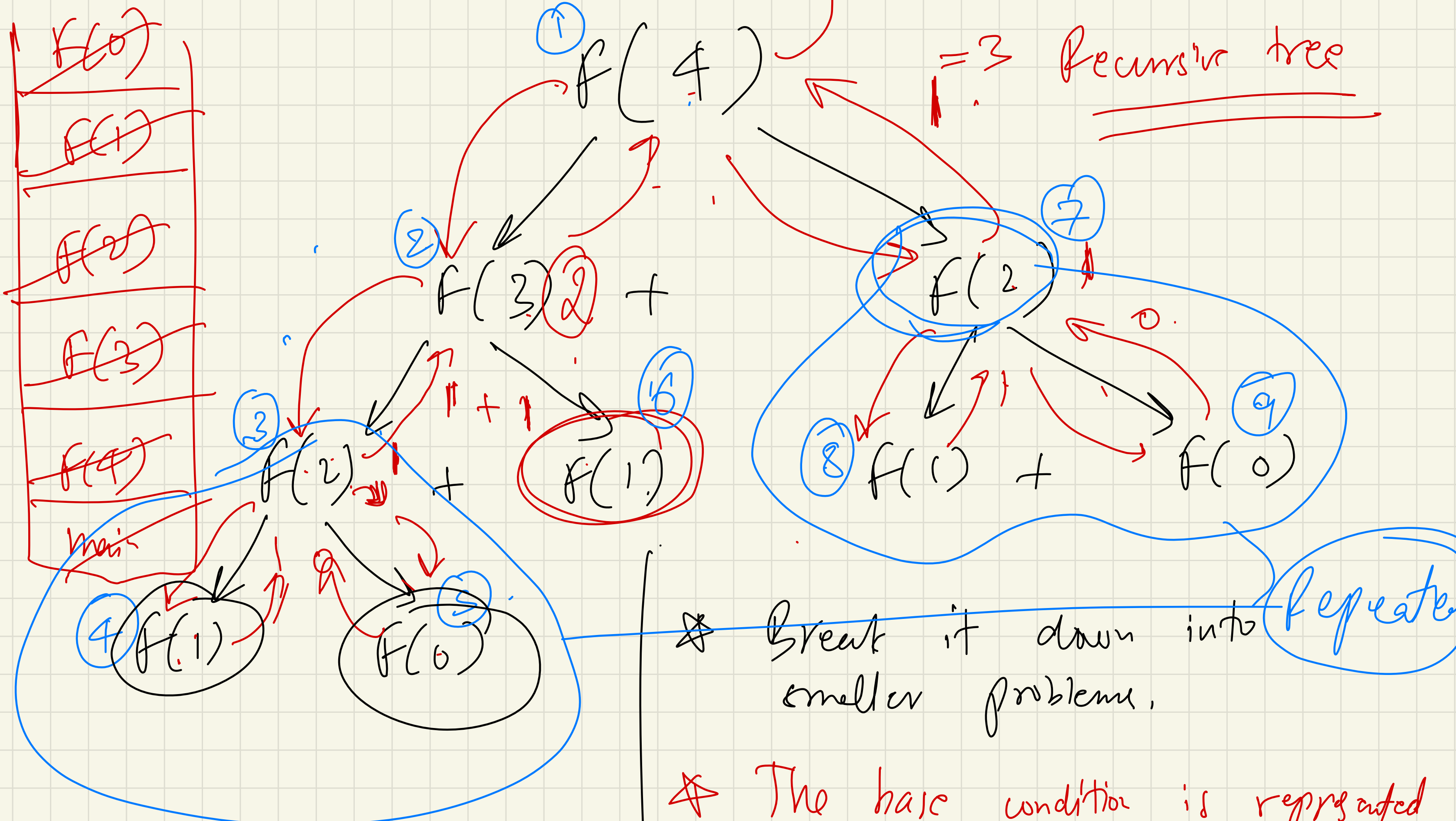
→ common sense

$Fibo(n-1) = Fibo(n-2) + Fibo(n-3)$

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

0th main

Recursive tree



* Break it down into smaller problems.

* The base condition is represented by answers we already have.

Note: How to understand & approach a problem:

WWI

1. Identify if you can break down problem into smaller problems
2. Write the recurrence relation if needed.
3. Draw the recursive tree
4. Print the tree:

In this case, we know that $F(0)=0$, $F(1)=1$

This is base condition!

WWU

* See the flow of functions, how they are getting in stuck.

- Identify & focus on left tree calls and right tree calls.
- Draw the tree and pointers again and again using pen & paper.
- Use a debugger to see the flow

(5) See how the values & what type of values (int, string etc.) are returned at each step. See where the function call will come out. In the end, you will come out of the main function.

(6) Subscribe to the channel, like, comment, share on socials. (VVVI)

Tip: Make sure to return the result of a function call of the return type.

VVVVVVI

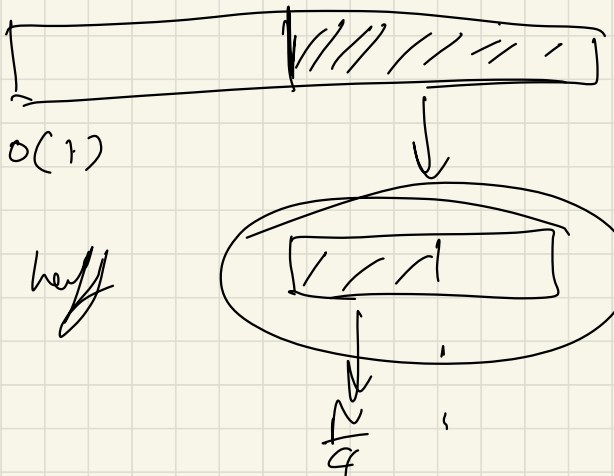
Variables : (1) Arguments
(2) Return Type
(3) Body of function.

will go in next func call

Q1 Binary Search with recursion

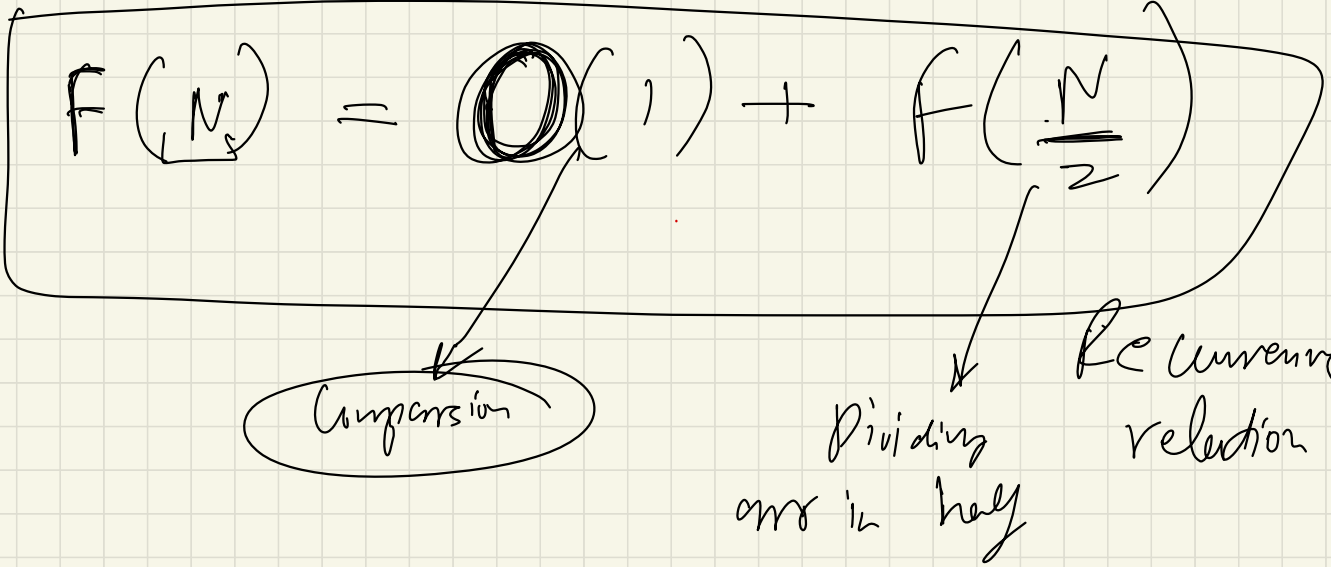
s, e

Divide to next call



(1) Comparing → O(1)

(2) Dividing into 2 halfs

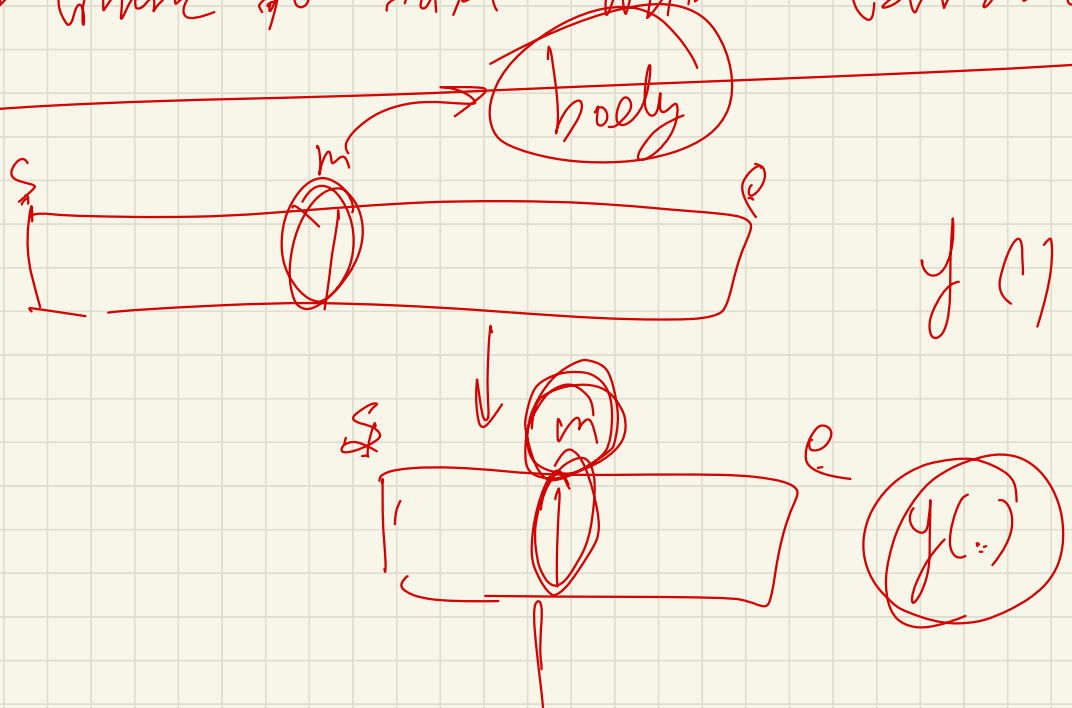


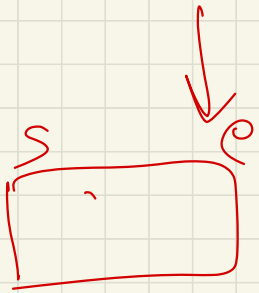
Types of recurrence relation:

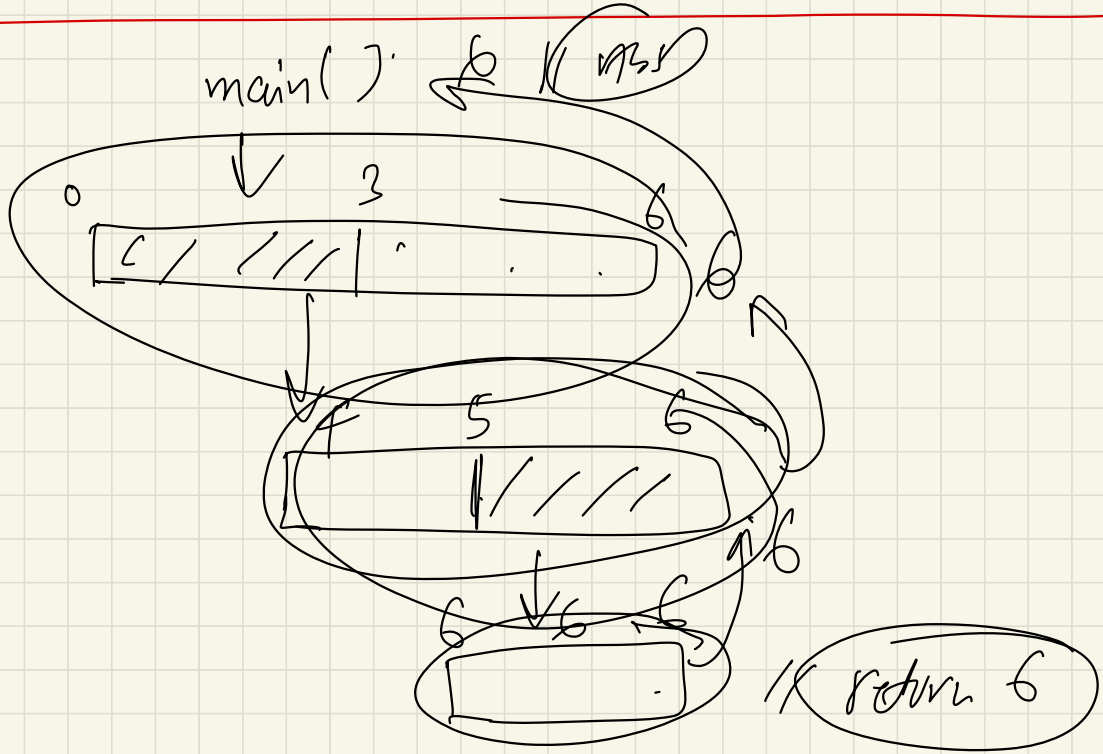
1. Linear recurrence relation → fib.
2. Divide & conquer rec relation → BS (reduced by a factor)

Tip: Do not overthink.

Continuation of where to take with variables!




s
e


main( ):
0
3
6
5
6
return 6