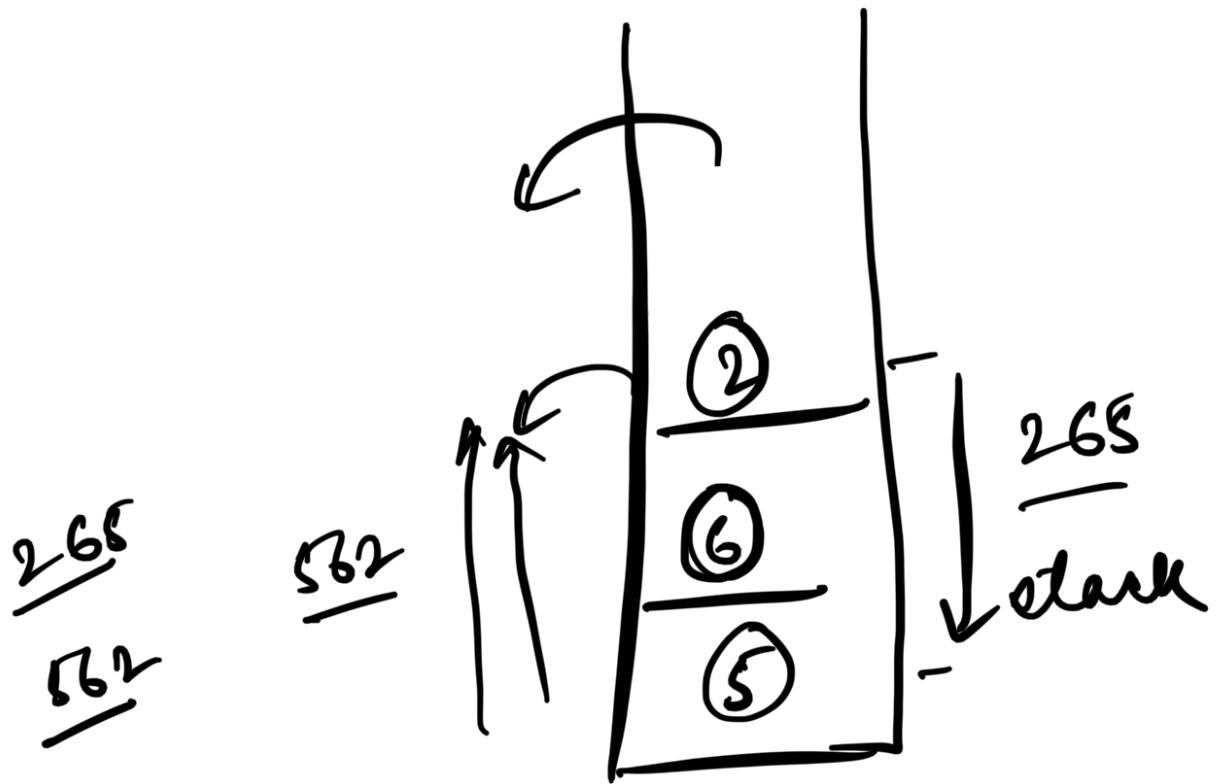


Recursion Problem Solving



$$\text{peek}() = 10, \text{pop}() = \underline{10}$$

$$\text{pop}() = 2$$

$$\text{peek}() = 6$$

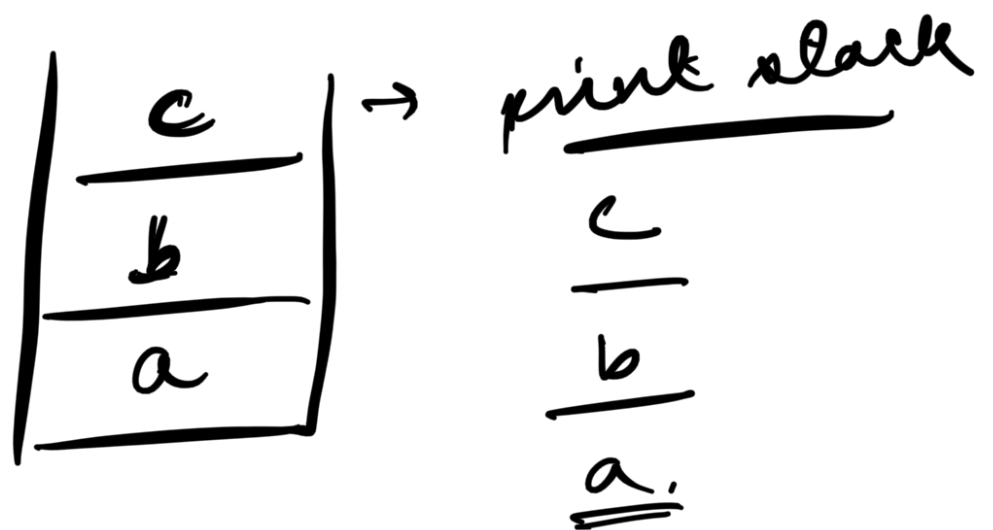
$$\text{peek}() = 6$$

$$\text{push}(2)$$

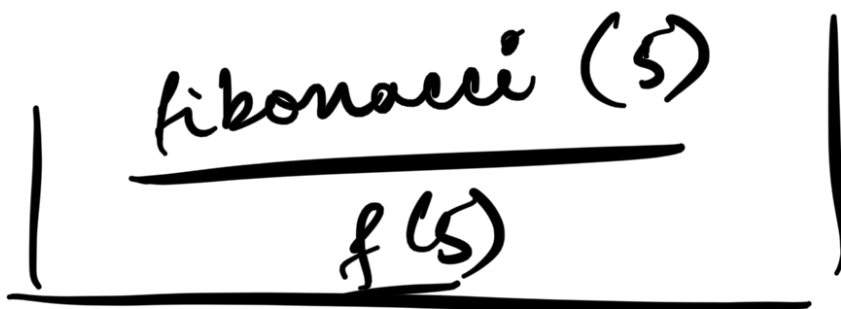
$$\text{showStack}() = (\text{bottom to top } 562)$$

↳ lfp to bottom
265

Call stack for function



Recursion Tree



— f(5)

fun f(0)

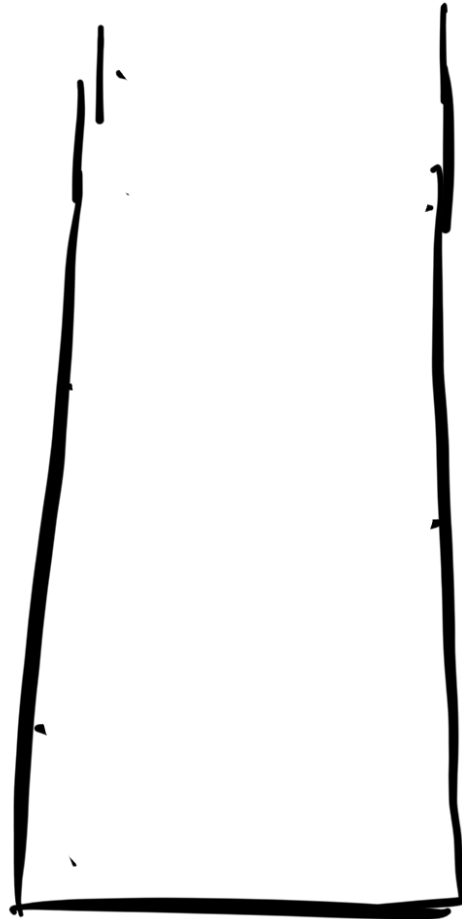
console.log(fibonacci(5));

5

Print Increasing

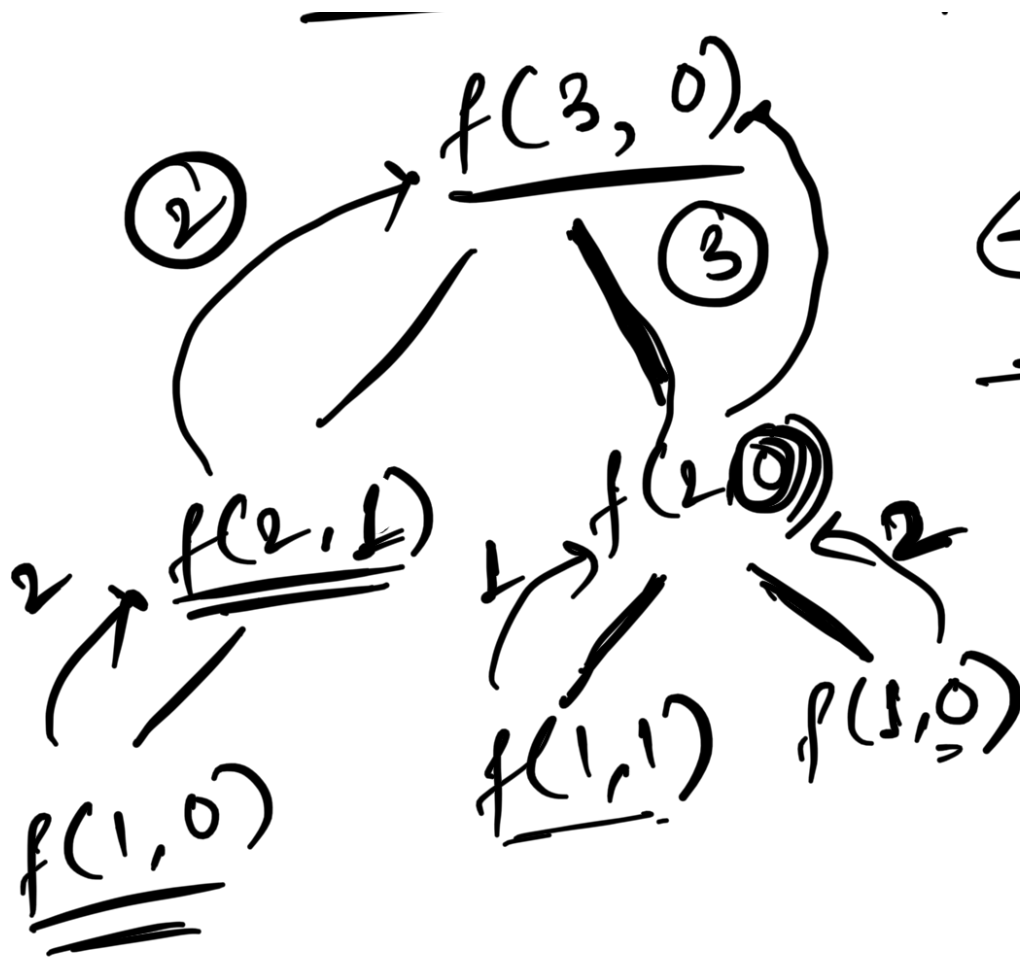
print(5);

1
2
3
4
5

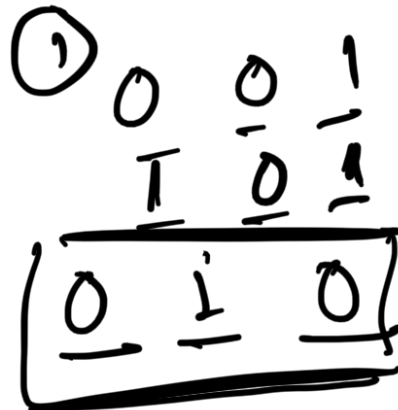
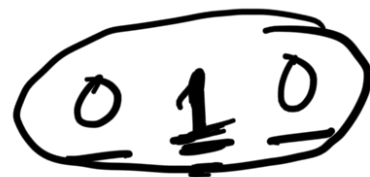
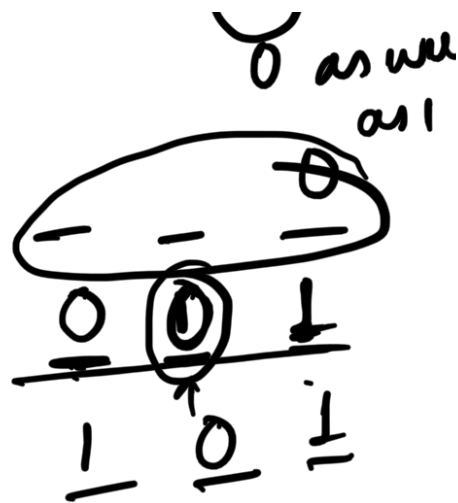


Calculate Zeros

-- (-)(0)



$$\underline{\underline{3+2=5}}$$



10

()

close < open {
add ("")

if (

open, close+1);

}
result.pop();

if (open < n) {
result.push('(');
parenthesis(result,
open+1, close);

if (close < open) {
result.push(')');
parenthesis(result,
open, close+1;
result.pop();

LLC

}