Recursion Basics

In This Lecture

1. Basics of Recursion
2. Recursion Stack
3. Linear Recursion and Tree Recursion
4. The sum of n Natural Numbers
5. Fibonacci Number

Basics of Recursion

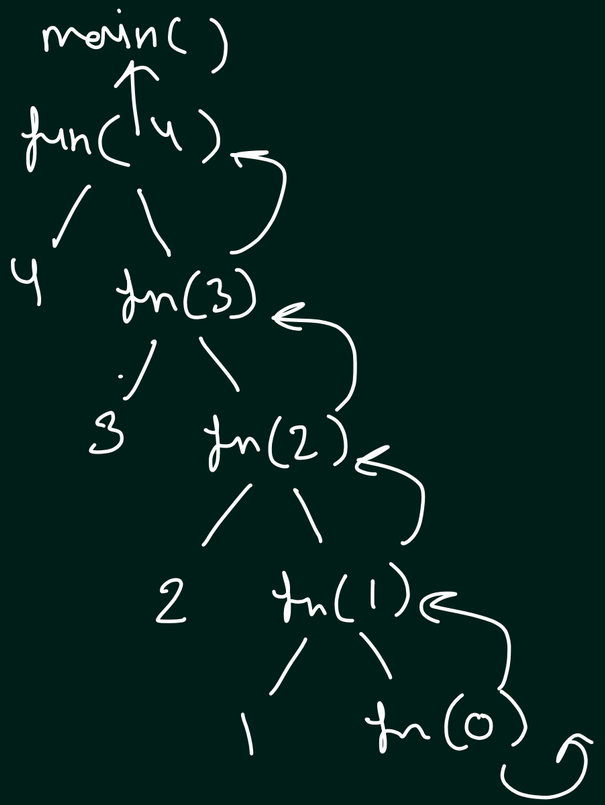
Recursion is the technique of making a function call itself. This technique provides a way to break complicated problems down into simple problems which are easier to solve.

n = 4 3 2 1 0

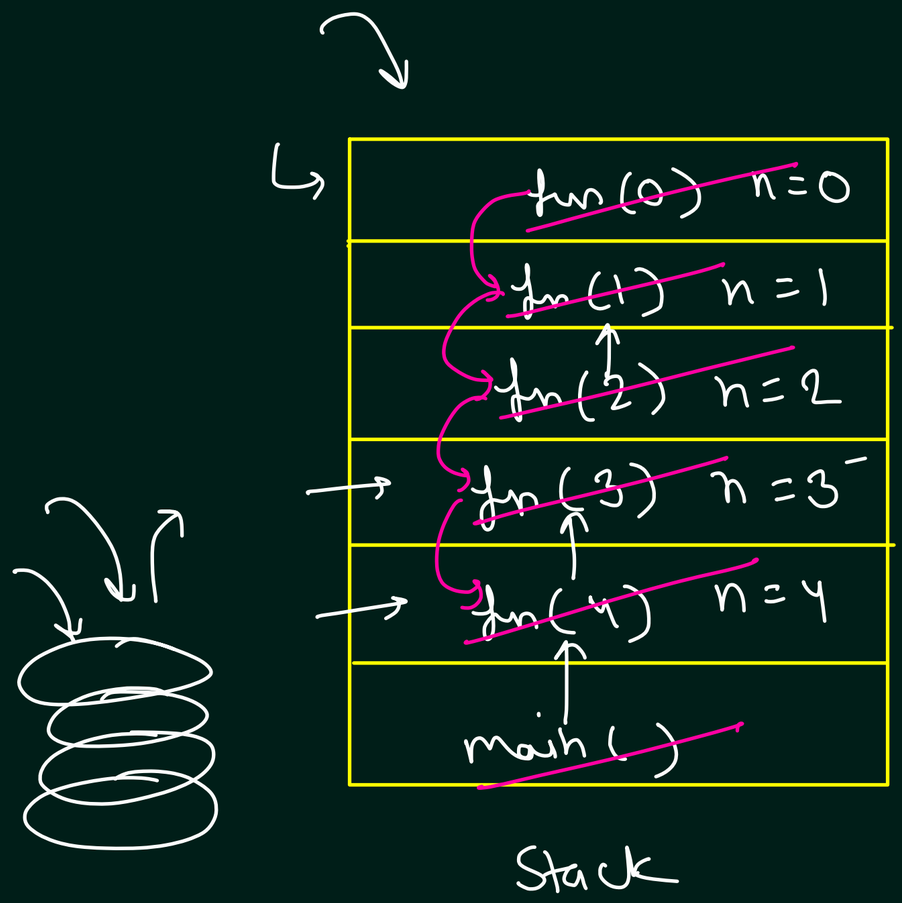
Base case

```
2 usages
static void fun(int n) {
    if(n > 0) {
        System.out.println(n);
        fun( n: n - 1);
    }
}
```

o/p : 4 3 2 1



Recursion Stack

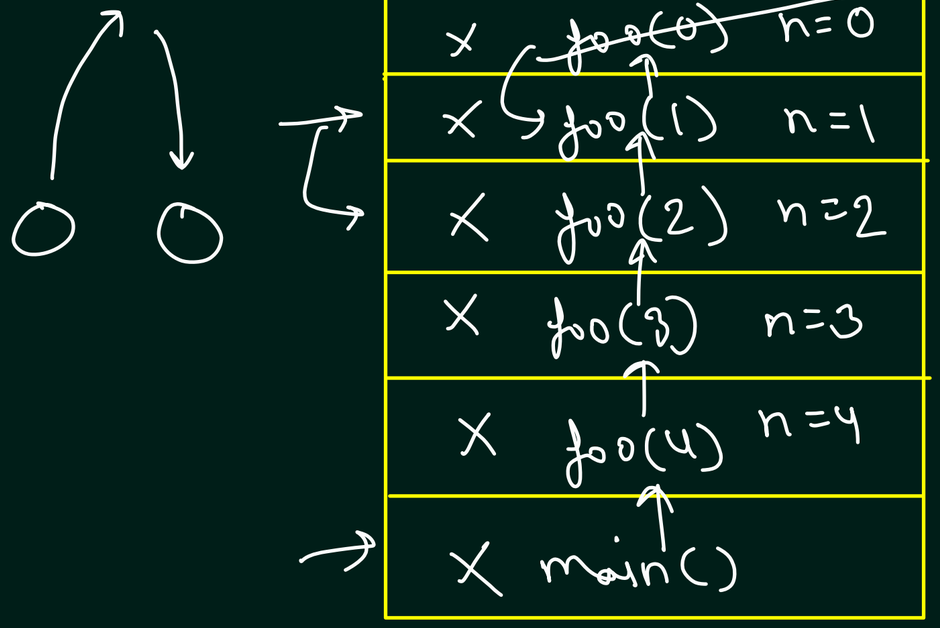



```
2 usages
static void fun(int n) {
    if(n > 0) {
        System.out.println(n);
        fun( n: n - 1);
    }
}
```

asc

op/ : 4 3 2 1

Recursion Stack

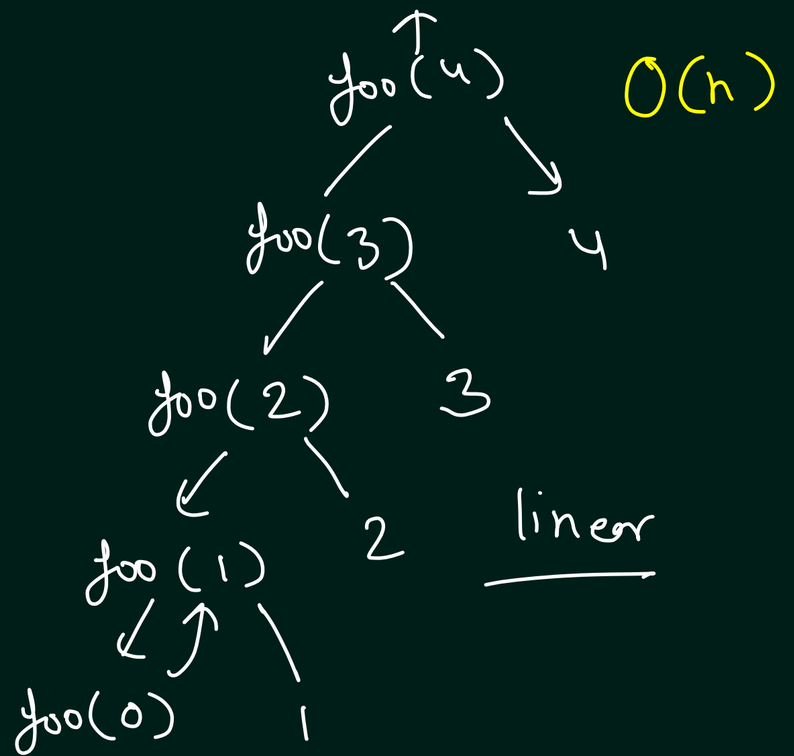


op: 1 2 3 4

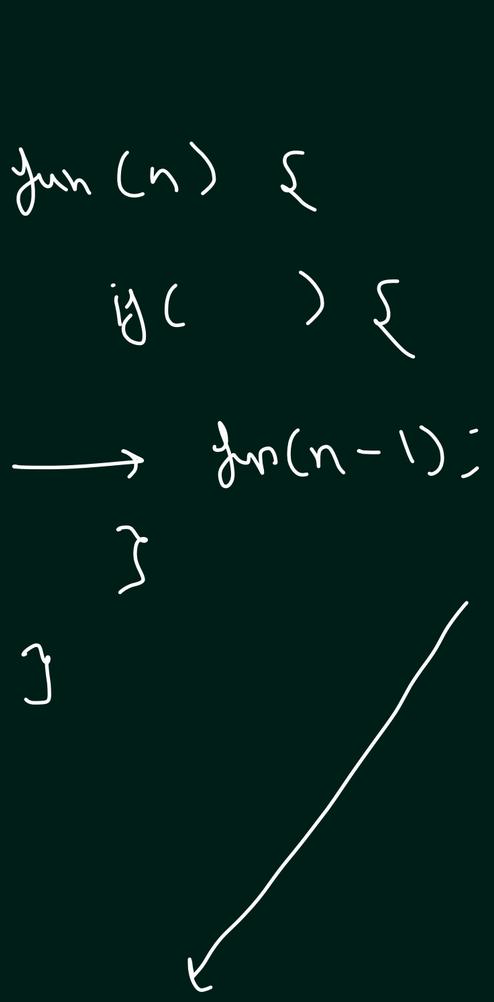
main()

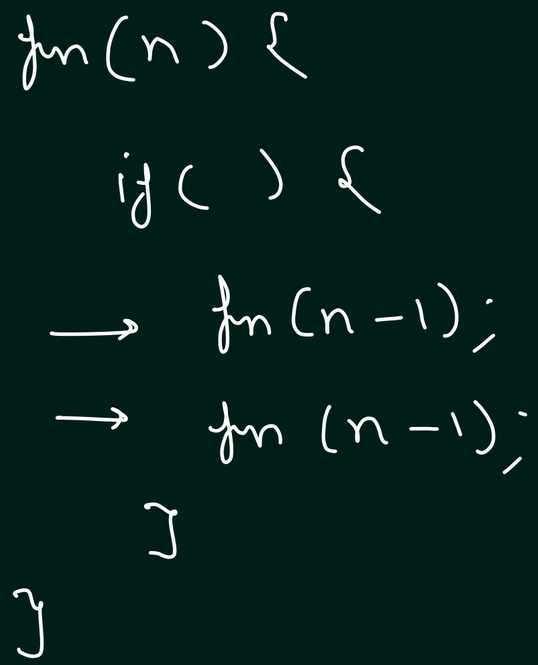
```
2 usages
static void foo(int n) {
    if(n > 0) {
        foo( n: n - 1);
        System.out.println(n);
    }
}
```

desc

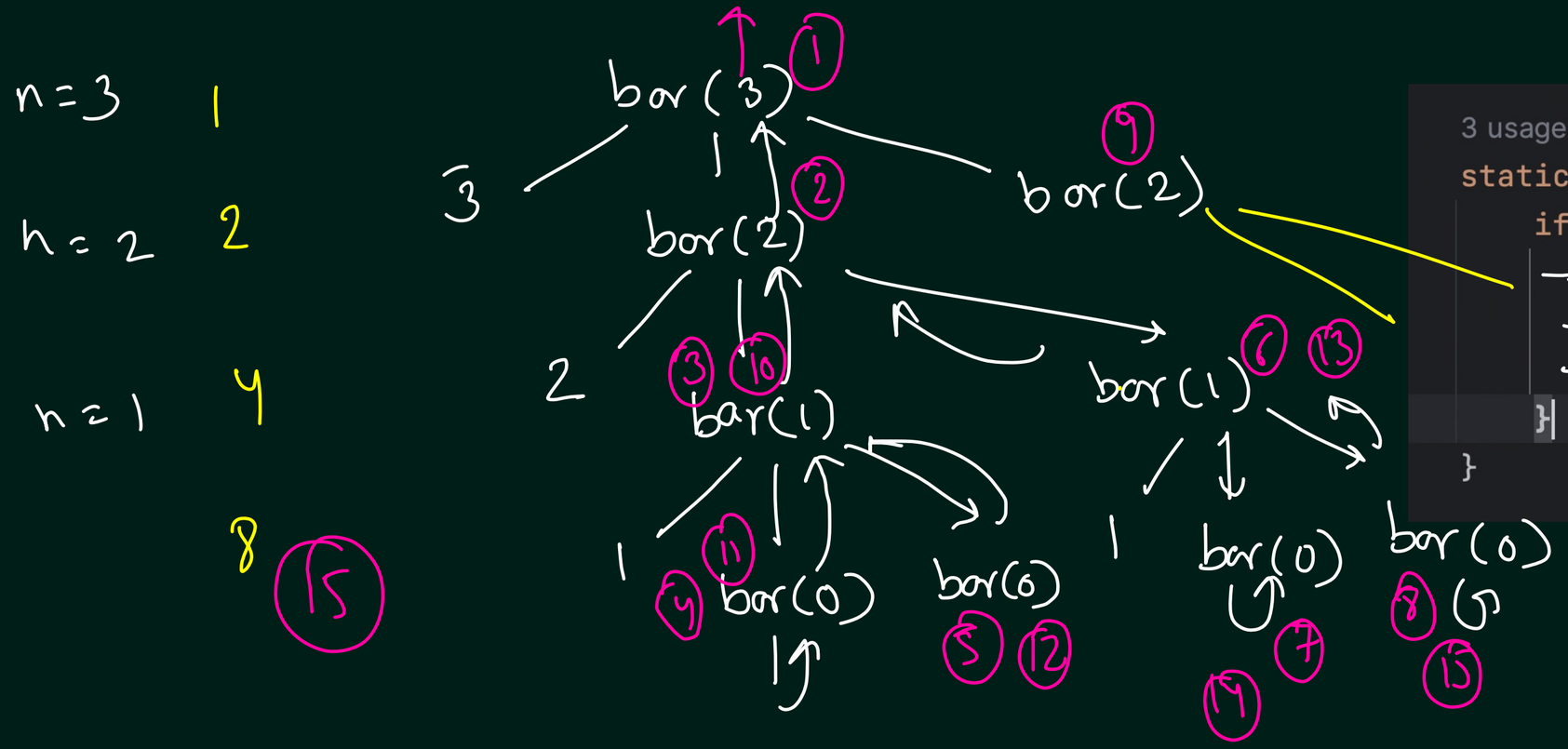


Linear Recursion & Tree Recursion





Linear Recursion & Tree Recursion




```
3 usages                    n=3
static void bar(int n) {
    if(n > 0) {
        System.out.println(n);
        bar( n: n-1);
        bar( n: n-1);    bar(2)
    }
}
```

$\hookrightarrow 2^{n+1} - 1$

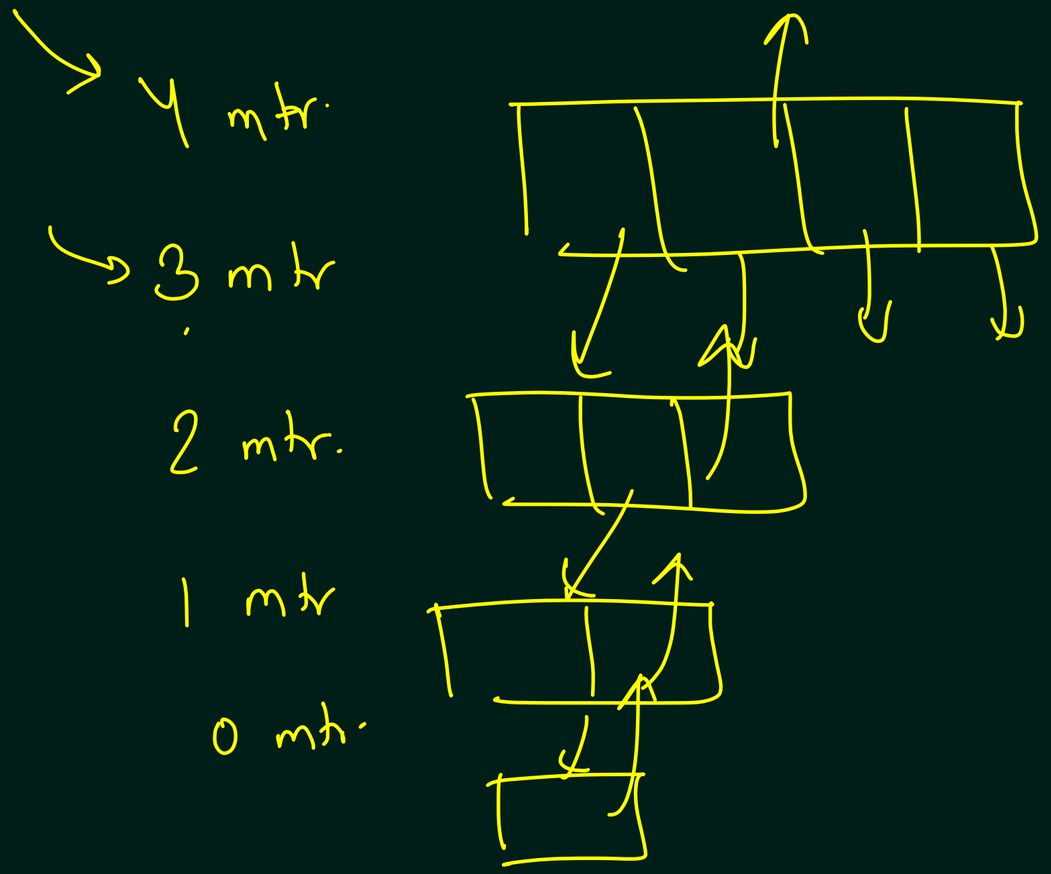
op: 3 2 1 1 2 1 1 $\quad 2^4 - 1 = 15$

$[2^0 + 2^1 + 2^2 + 2^3 \cdots 2^n = 2^{n+1} - 1] \longrightarrow \boxed{O(2^n)}$

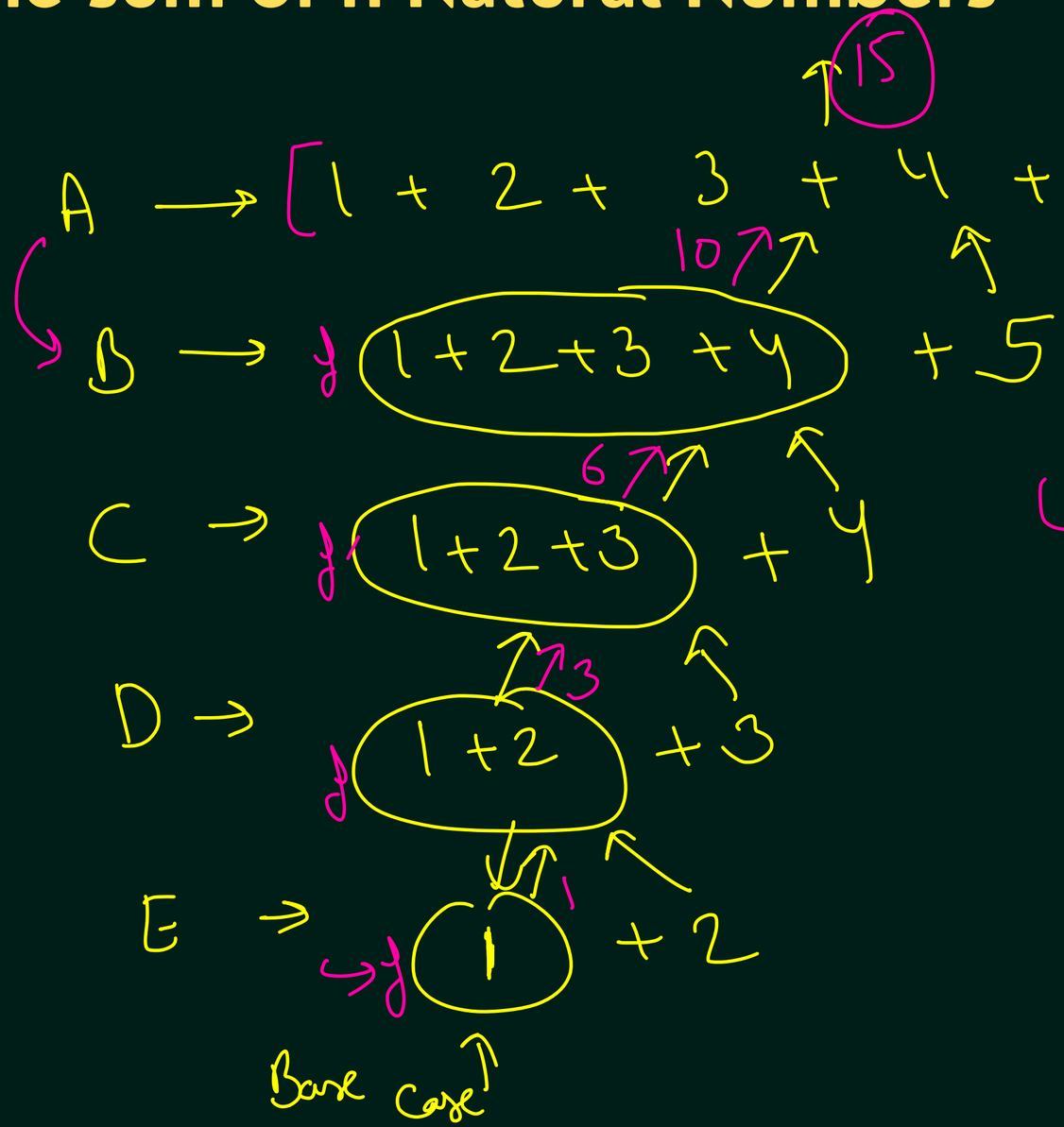
3 Steps To Solve Recursion Problem

- Step 1: Find the Base Case
- Step 2: Find the relation between the Problem and the Sub Problem

Step 3: Generalise the relation found in the step 2



The sum of n Natural Numbers



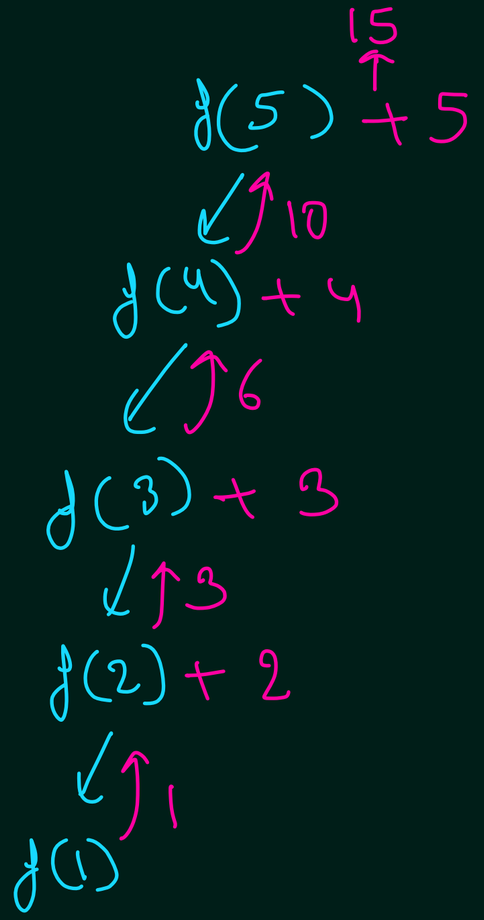
n

→ n == 1 (Base Condition)

↳ [f(n) = f(n-1) + n]

↳ [f(5) = f(4) + 5]

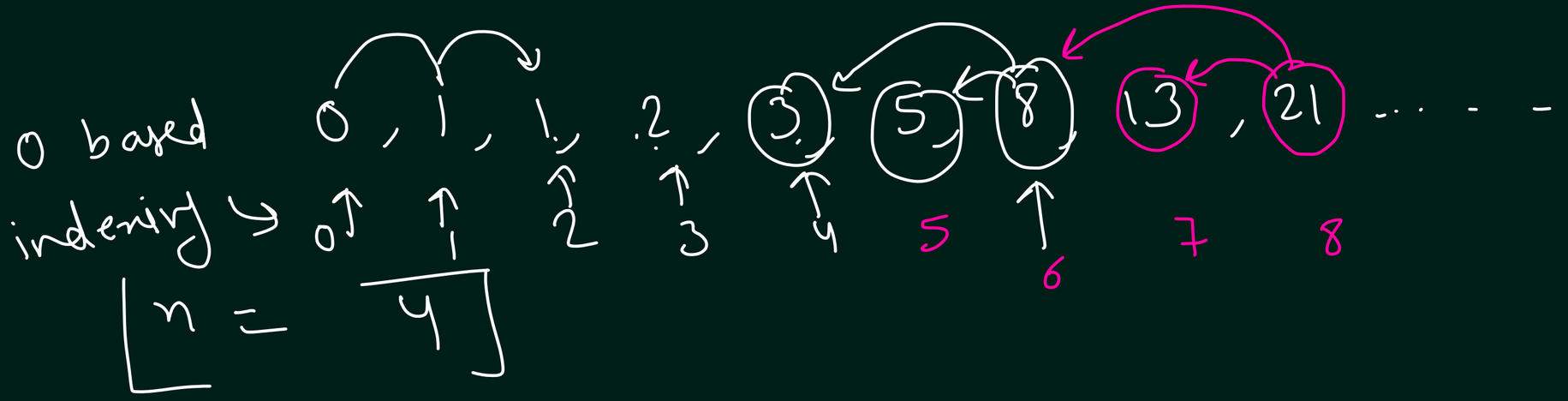
The sum of n Natural Numbers

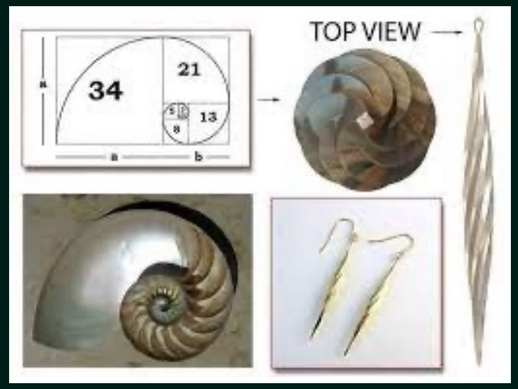



```java
System.out.println(sumOfN( n: 5));
}

2 usages
static int sumOfN(int n) {
    if(n == 1) return 1;
    return sumOfN( n: n-1) + n;
}
```

Fibonacci Number



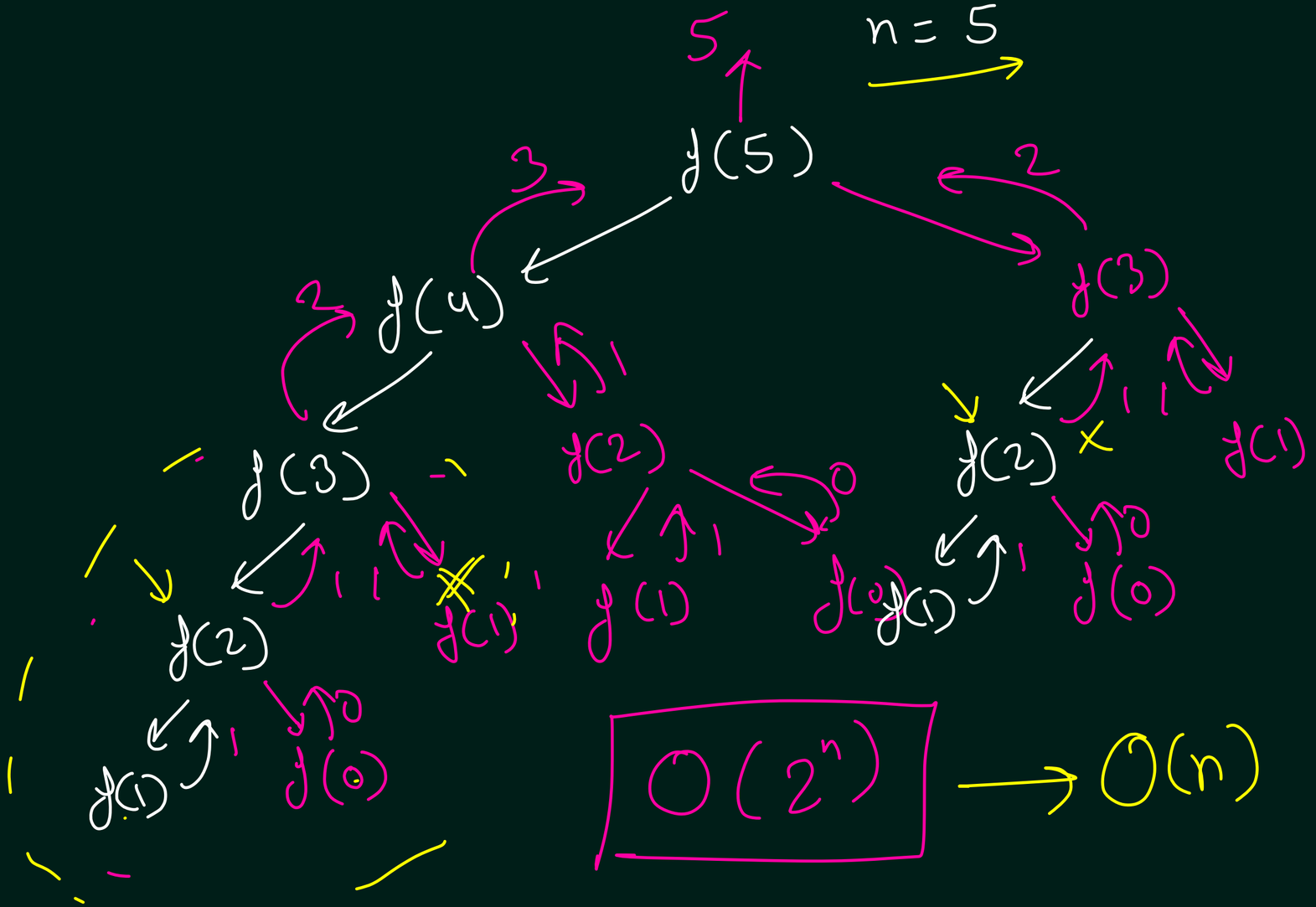


$$f(n) = f(n-1) + f(n-2)$$

$$f(8) = f(7) + f(6)$$

$$= 13 + 8 = 21$$

$$f(2) = f(1) + f(0) = 1$$




```
3 usages                                    5
static int fibOfN(int n) {
    if(n <= 1) return n;
    return fibOfN( n: n-1) + fibOfN( n: n-2);
}
```

f(4) = 3 f(3) = 2

memoization