# MASTER
# RECURSION

CPMSOC

Bharat Singla

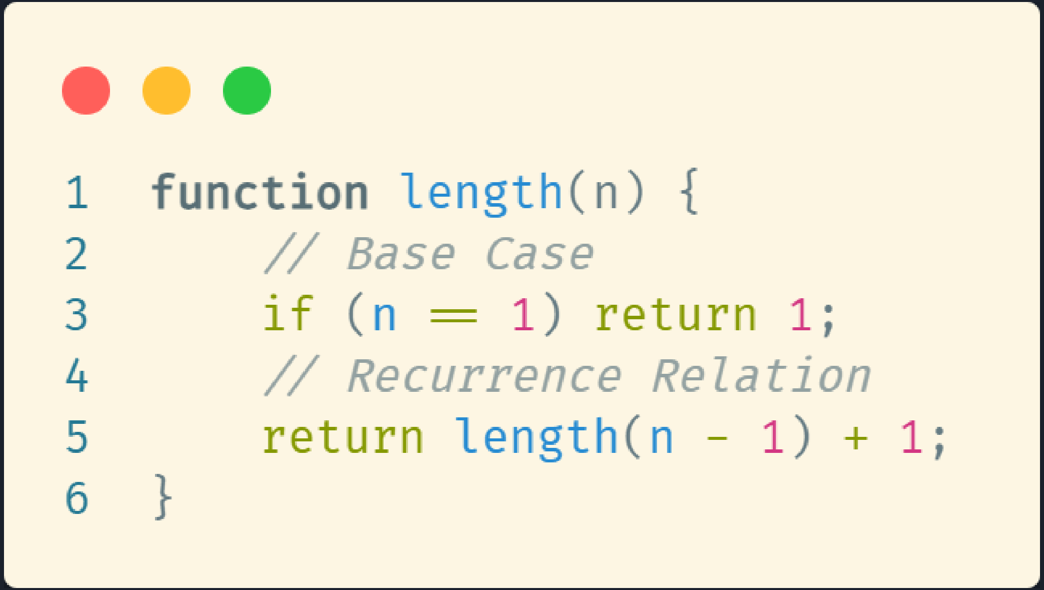
Yi Heng

# Length of Queue?

# Recursion

- ➔ Decompose the problem into similar sub-problems
- ➔ Solve the sub-problems (magically!) using recursion
- ➔ Merge the solutions to get the final soln

# Recursion

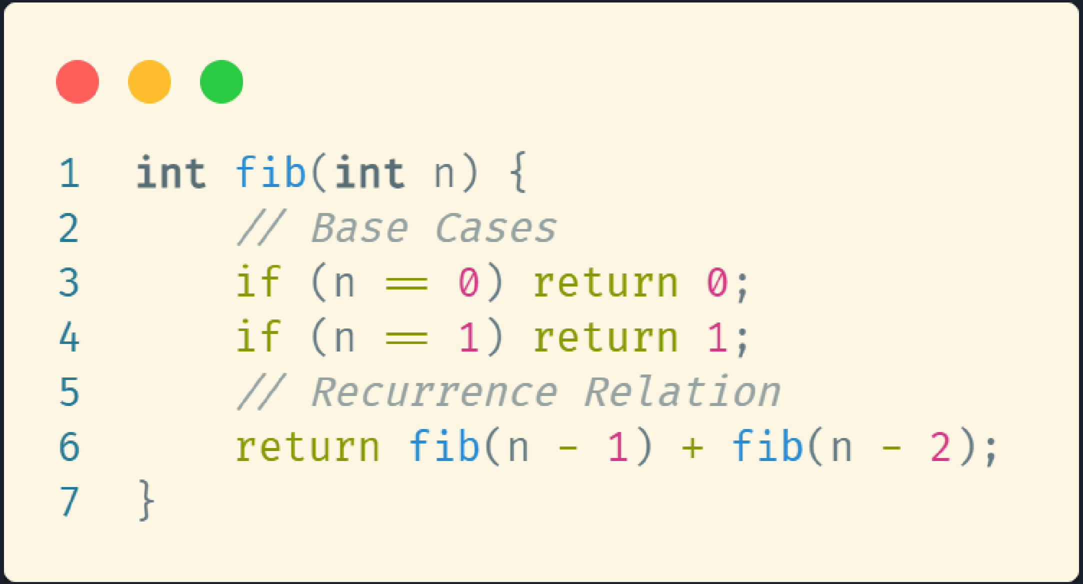


```
function length(n) {
    // Base Case
    if (n == 1) return 1;
    // Recurrence Relation
    return length(n - 1) + 1;
}
```

# Fibonacci!

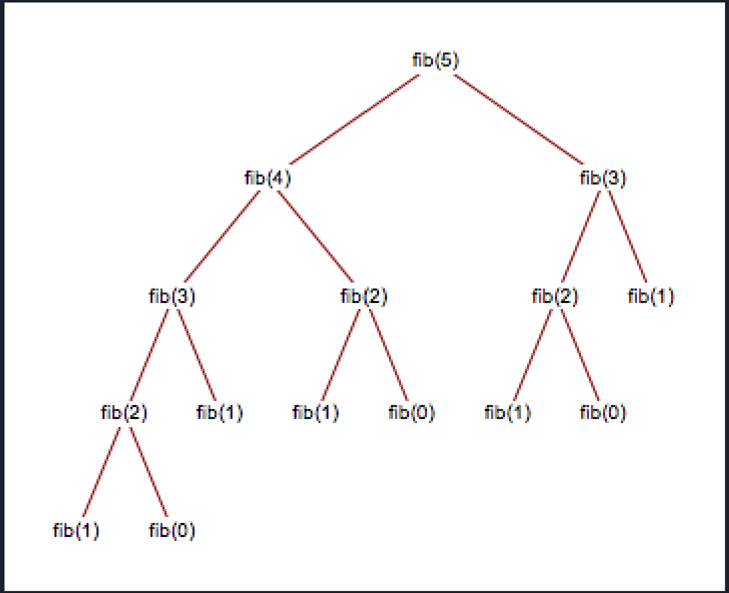
- Each term of the sequence is defined as the sum of previous 2 terms
- Fib(0) = 0, Fib(1) = 1
- Fib(n) = Fib(n-1) + Fib(n-2)
- 0, 1, 1, 2, 3, 5, 8…

# Fibonacci!



```
int fib(int n) {
    // Base Cases
    if (n == 0) return 0;
    if (n == 1) return 1;
    // Recurrence Relation
    return fib(n - 1) + fib(n - 2);
}
```

## Recursion Tree



## Recursion Stack

fib(1)

fib(2)

fib(3)

fib(4)

fib(5)

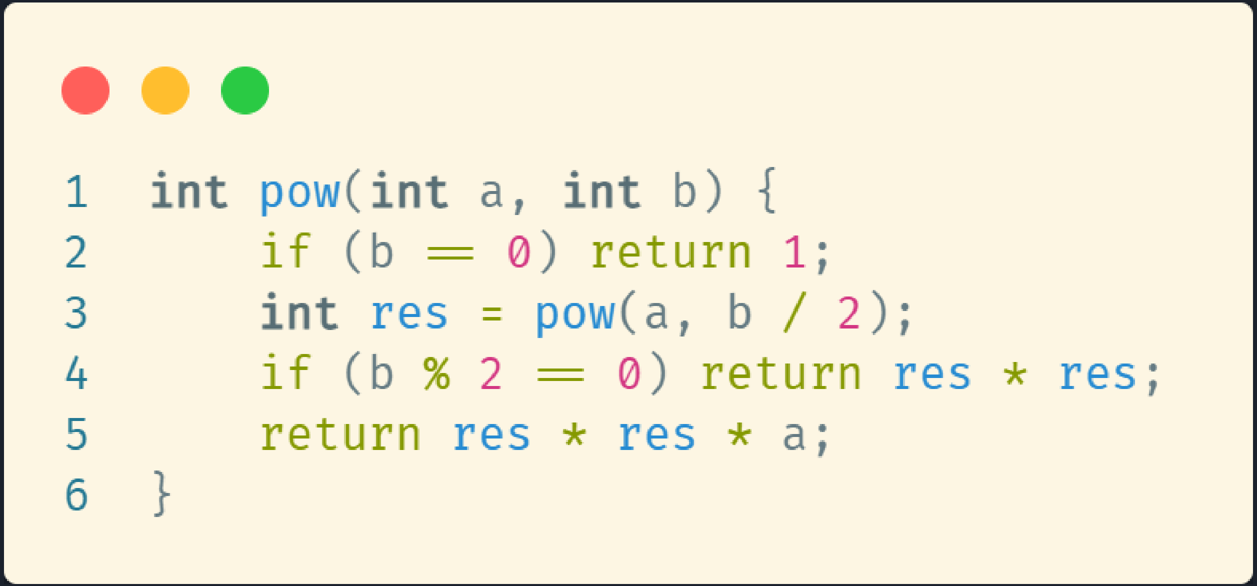
# Time & Space Complexity?

- ➔ Time complexity: O($2^n$)
- ➔ Space complexity: O(n)

# Exponentiation

- ➔ Find $a^b$ using recursion!
- ➔ Time complexity: O(log(b))
- ➔ Space complexity: O(log(b))

# Exponentiation



```
int pow(int a, int b) {
    if (b == 0) return 1;
    int res = pow(a, b / 2);
    if (b % 2 == 0) return res * res;
    return res * res * a;
}
```

# More Recursion!?

- ➔ Printing nums from 1 to n
- ➔ Factorial
- ➔ Sum of node values of a linked list
- ➔ Count of a letter in a string
- ➔ Subsets of an array

# That’s all folks!

Arc

CLUBS

SUPPORTED BY Arc
INDEPENDENTLY RUN