CSC 240
Computer Graphics

Sara Mathieson
Fall 2016
Smith College

HW 2 due Tuesday **(tomorrow)**

Outline: 9/26

- HW 1 demos
- Recap flood fill
- Sweep fill
- Begin: **transformations**

Turn in slips if you went to **Spinelli Center** review session

Office Hour rooms finalized

Monday 4-5pm (Ford 015)
Tuesday 4-5pm (Ford 346)

Laptops have arrived!
(info on Piazza)

Homework 1

Homework 1 Solution

Change paintbrush color first

```
function line(x1,y1,x2,y2,color) {
    graphics.fillStyle = color;

    // x is changing faster than (or equal to y)
    if (Math.abs(x1-x2) >= Math.abs(y1-y2)) {

        var m = (y2-y1)/(x2-x1); // compute slope

        // start from the minimum of x1 and x2
        var y = y1;
        if (x1>x2) {
            y = y2;
        }

        // loop over x
        for (var x = Math.min(x1,x2); x <= Math.max(x1,x2); x++) {
            graphics.fillRect(x, y, 1, 1);
            y += m;
        }

    // y is changing faster than x
    } else {

        var m_inverse = (x2-x1)/(y2-y1); // compute inverse of slope

        // start from the minimum of y1 and y2
        var x = x1;
        if (y1>y2) {
            x = x2;
        }

        // loop over y
        for (var y = Math.min(y1,y2); y <= Math.max(y1,y2); y++) {
            graphics.fillRect(x,y,1,1);
            x += m_inverse;
        }
    }
}
```

Homework 1 Solution

Change paintbrush color first

x changing faster than y, loop over x

```
function line(x1,y1,x2,y2,color) {
    graphics.fillStyle = color;

    // x is changing faster than (or equal to y)
    if (Math.abs(x1-x2) >= Math.abs(y1-y2)) {

        var m = (y2-y1)/(x2-x1); // compute slope

        // start from the minimum of x1 and x2
        var y = y1;
        if (x1>x2) {
            y = y2;
        }

        // loop over x
        for (var x = Math.min(x1,x2); x <= Math.max(x1,x2); x++) {
            graphics.fillRect(x, y, 1, 1);
            y += m;
        }

    // y is changing faster than x
    } else {

        var m_inverse = (x2-x1)/(y2-y1); // compute inverse of slope

        // start from the minimum of y1 and y2
        var x = x1;
        if (y1>y2) {
            x = x2;
        }

        // loop over y
        for (var y = Math.min(y1,y2); y <= Math.max(y1,y2); y++) {
            graphics.fillRect(x,y,1,1);
            x += m_inverse;
        }
    }
}
```

Homework 1 Solution

Change paintbrush color first

x changing faster than y, loop over x

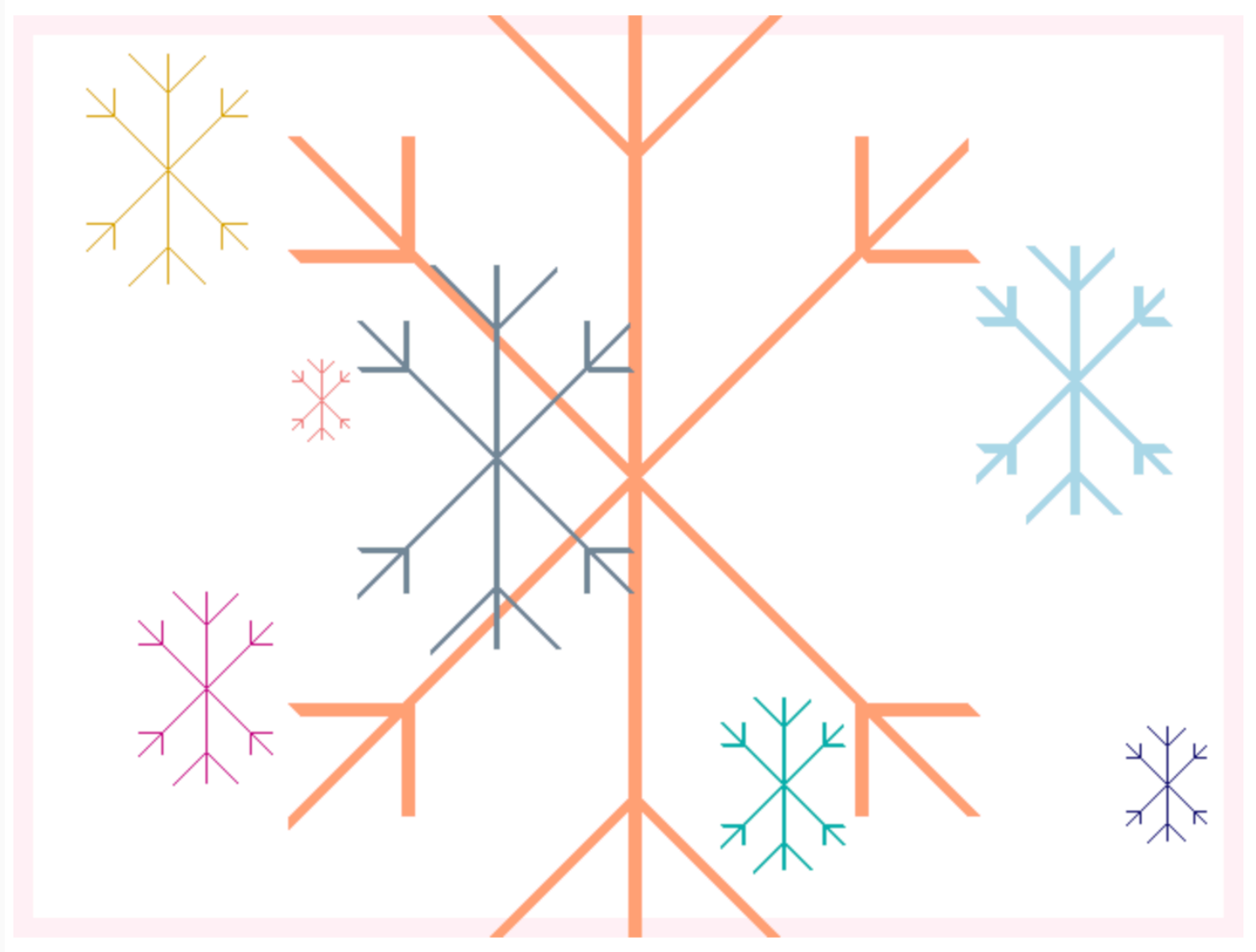
y changing faster than x, loop over y

```
function line(x1,y1,x2,y2,color) {
    graphics.fillStyle = color;

    // x is changing faster than (or equal to y)
    if (Math.abs(x1-x2) >= Math.abs(y1-y2)) {

        var m = (y2-y1)/(x2-x1); // compute slope

        // start from the minimum of x1 and x2
        var y = y1;
        if (x1>x2) {
            y = y2;
        }

        // loop over x
        for (var x = Math.min(x1,x2); x <= Math.max(x1,x2); x++) {
            graphics.fillRect(x, y, 1, 1);
            y += m;
        }

    // y is changing faster than x
    } else {

        var m_inverse = (x2-x1)/(y2-y1); // compute inverse of slope

        // start from the minimum of y1 and y2
        var x = x1;
        if (y1>y2) {
            x = x2;
        }

        // loop over y
        for (var y = Math.min(y1,y2); y <= Math.max(y1,y2); y++) {
            graphics.fillRect(x,y,1,1);
            x += m_inverse;
        }
    }
}
```

Homework 1

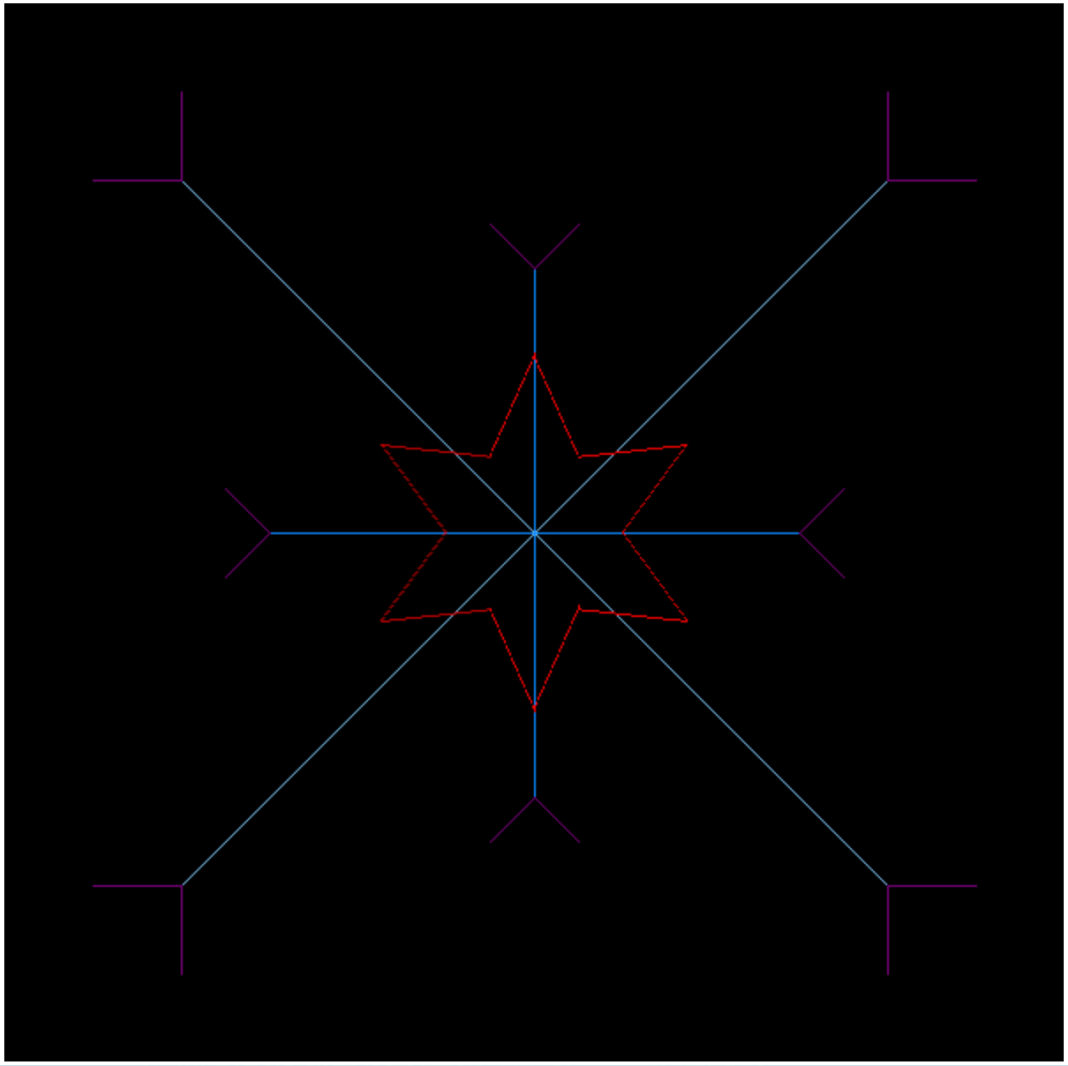
- There is code on the web for many of these algorithms… write without looking at this code.
- Think about all cases, test with all inputs.

Homework 1 Examples



Credit: Leticia

Homework 1 Examples



Credit: Mara

HW 1 Web Examples

Revisit Recursion

Fibonacci Function Stack

fib(5)

```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → var out1 = fib(n-1);
Line B → var out2 = fib(n-2);
Line C → return out1+out2;

A

fib(5)

current line

Function Stack

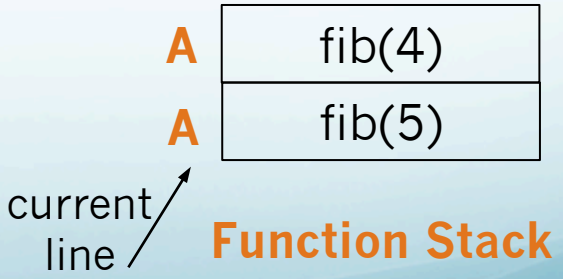
Fibonacci Function Stack

fib(5)

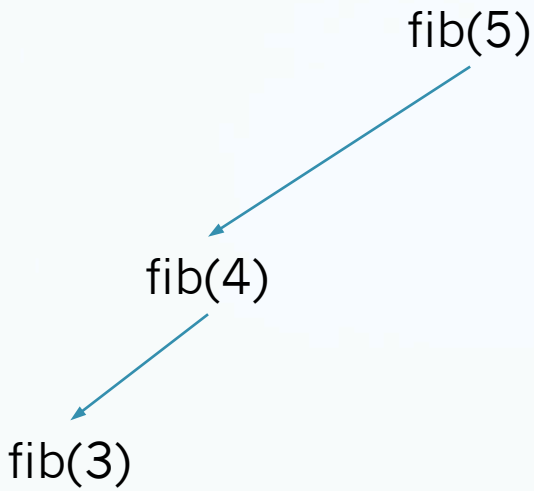
fib(4)

```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → var out1 = fib(n-1);
Line B → var out2 = fib(n-2);
Line C → return out1+out2;

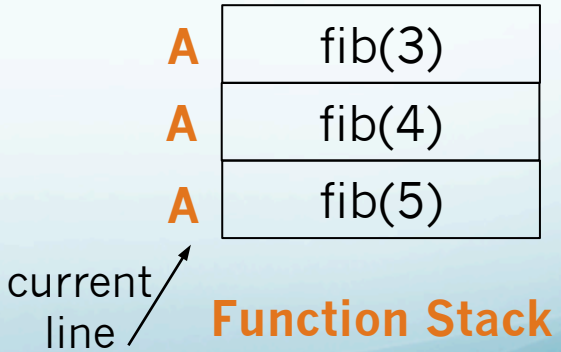


Fibonacci Function Stack

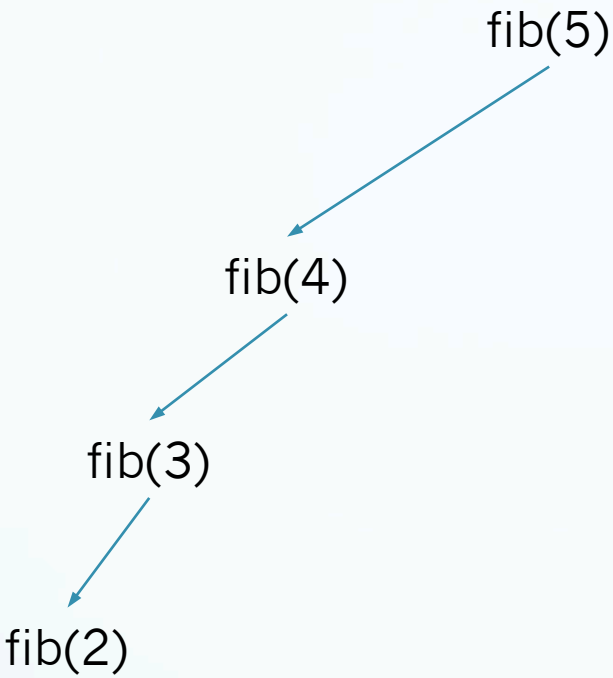



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

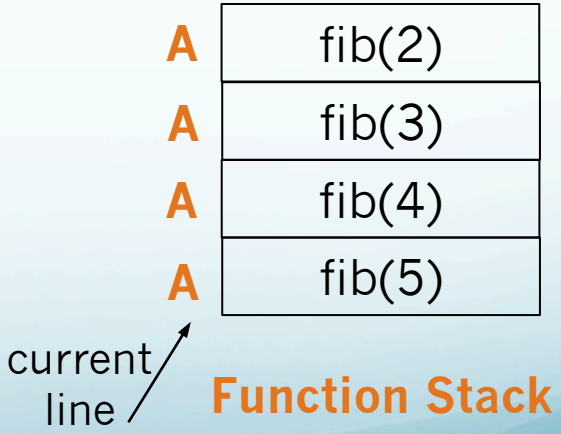


Fibonacci Function Stack

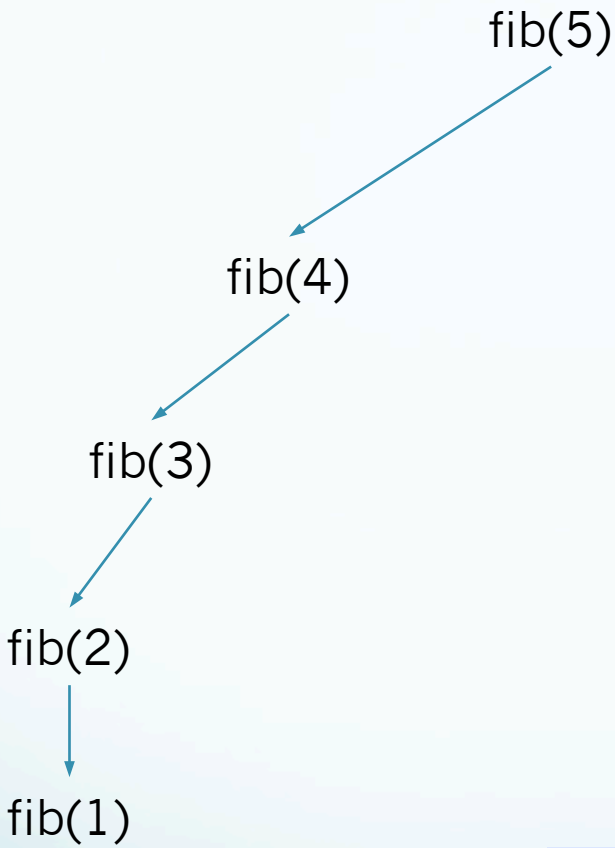



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

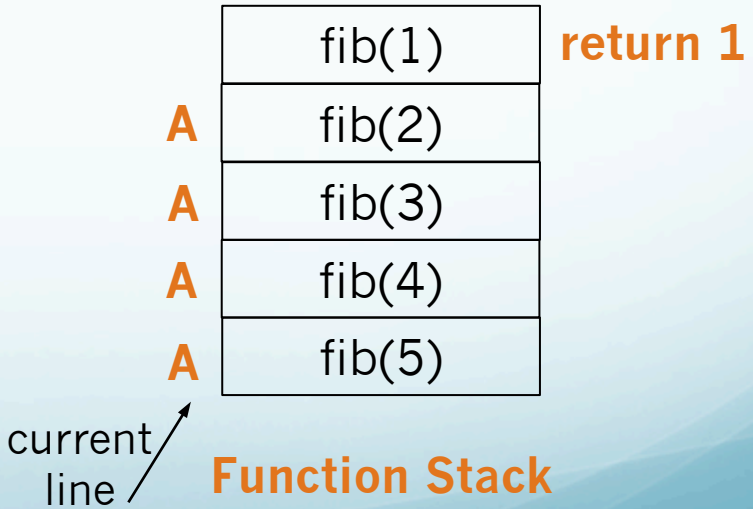


Fibonacci Function Stack

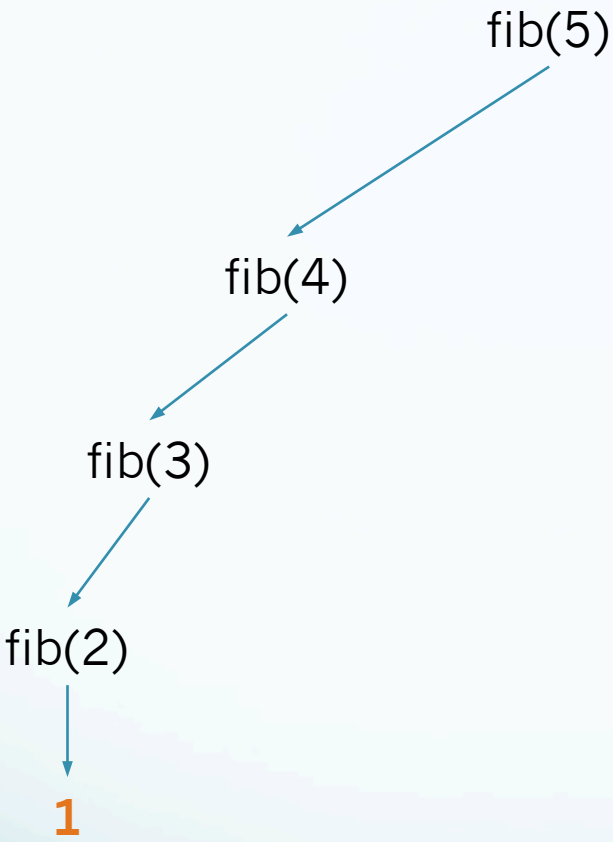



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

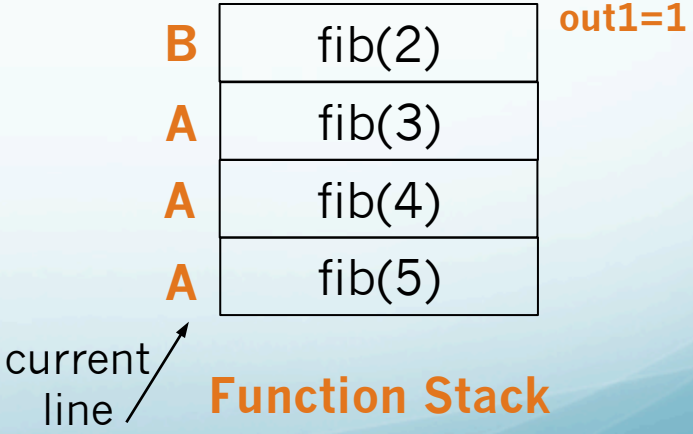
Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`



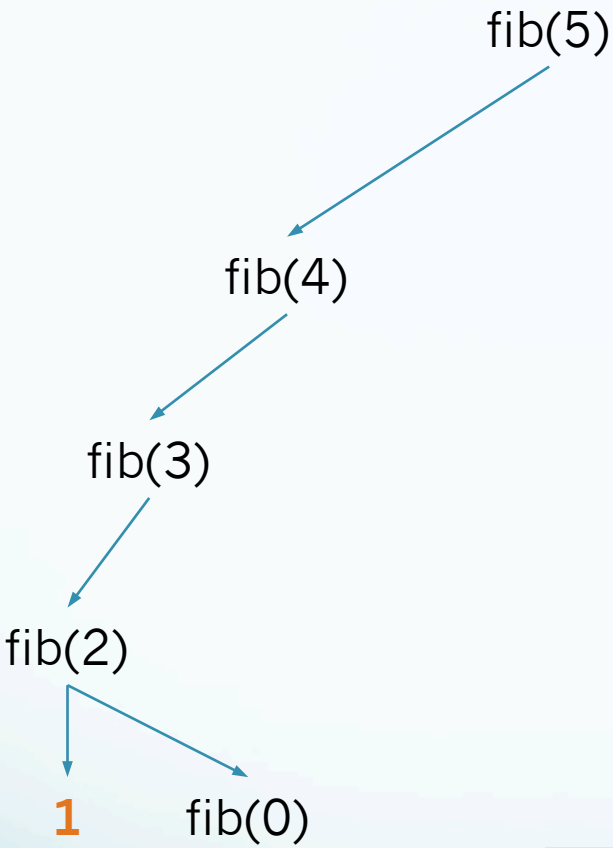
Fibonacci Function Stack




```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
Line A  var out1 = fib(n-1);
Line B  var out2 = fib(n-2);
Line C  return out1+out2;
}
```

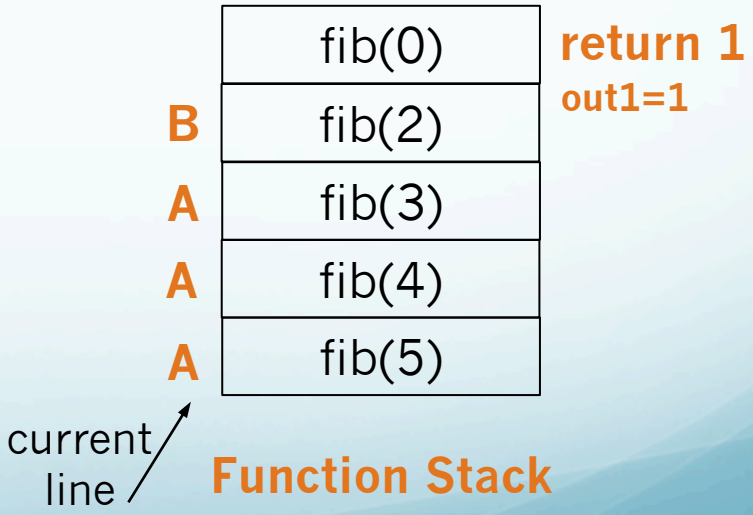


Fibonacci Function Stack

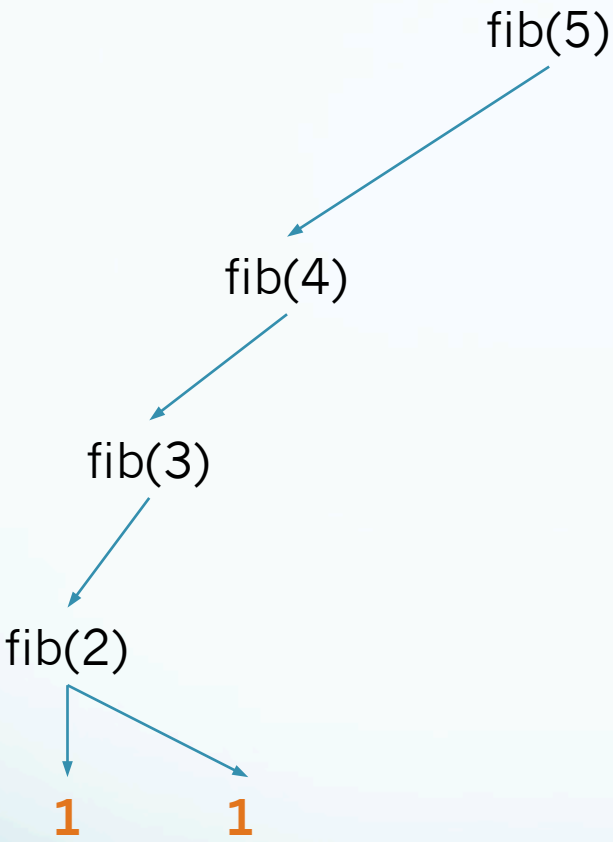



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

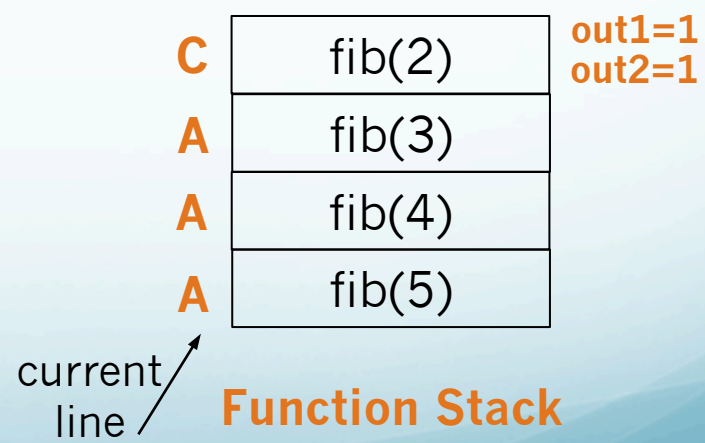


Fibonacci Function Stack

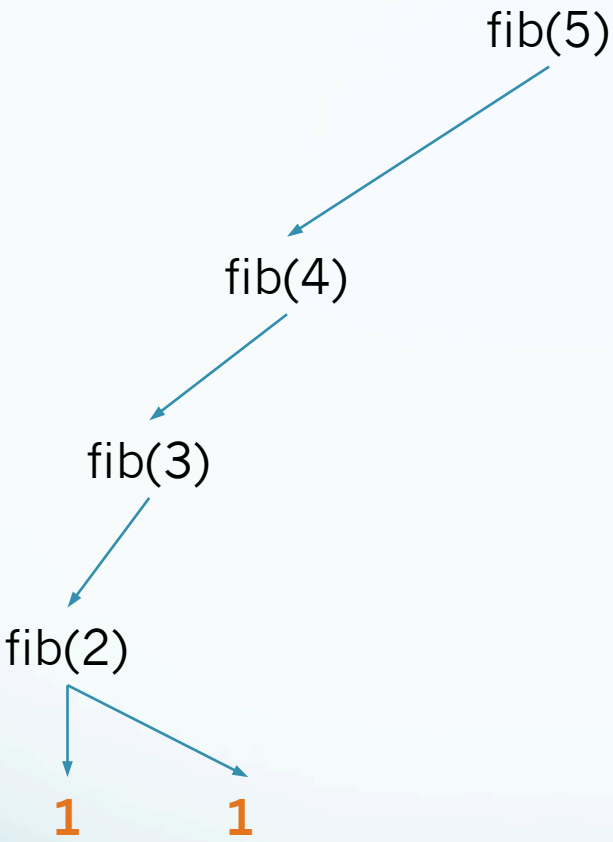



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

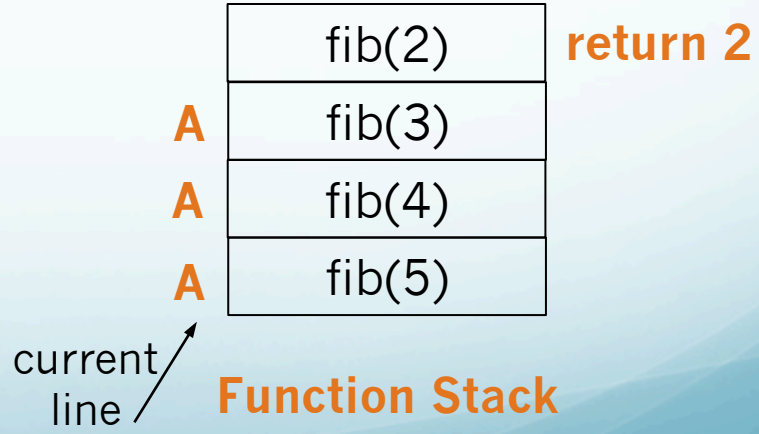


Fibonacci Function Stack

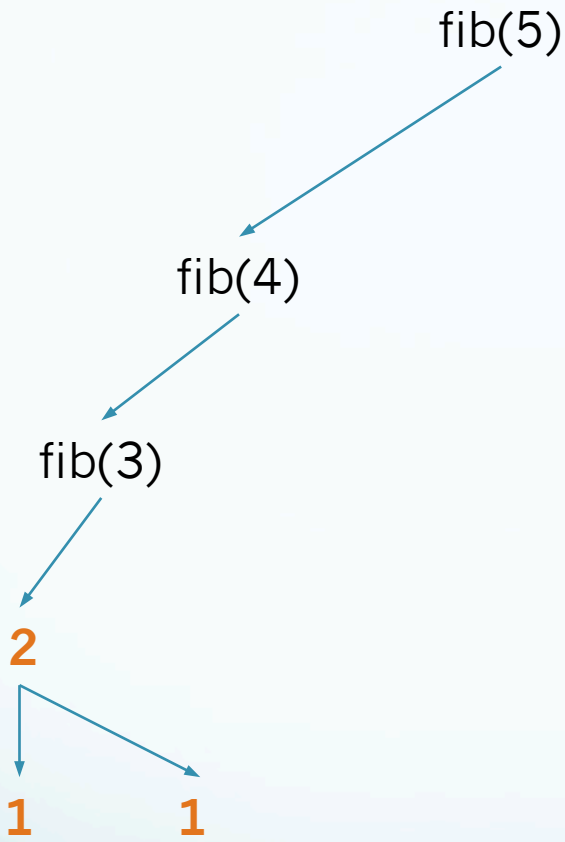



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

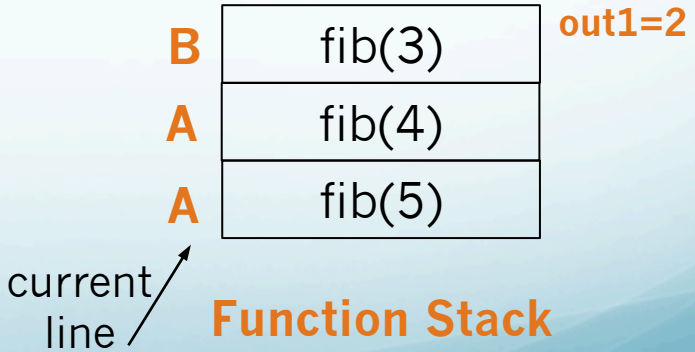


Fibonacci Function Stack

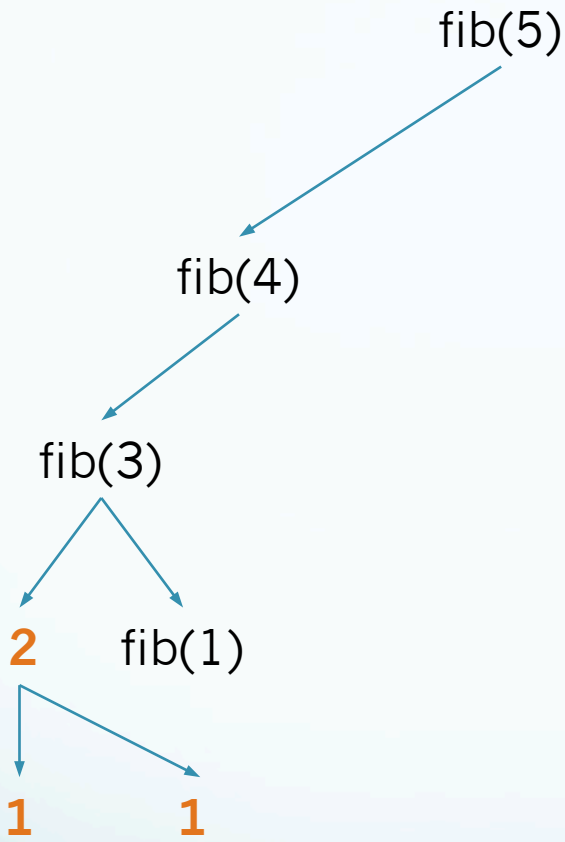



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

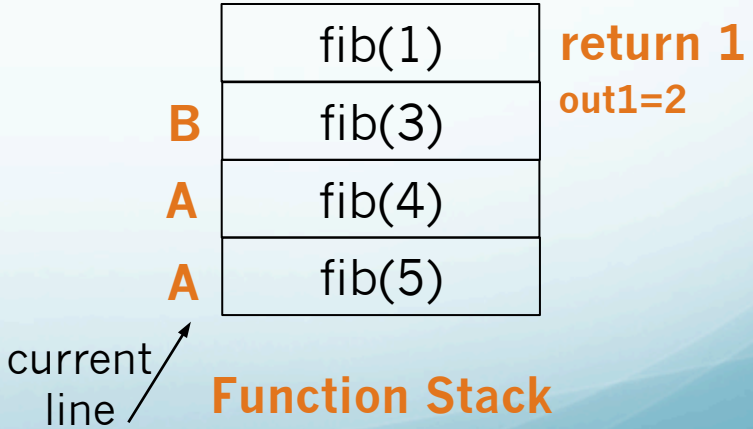


Fibonacci Function Stack

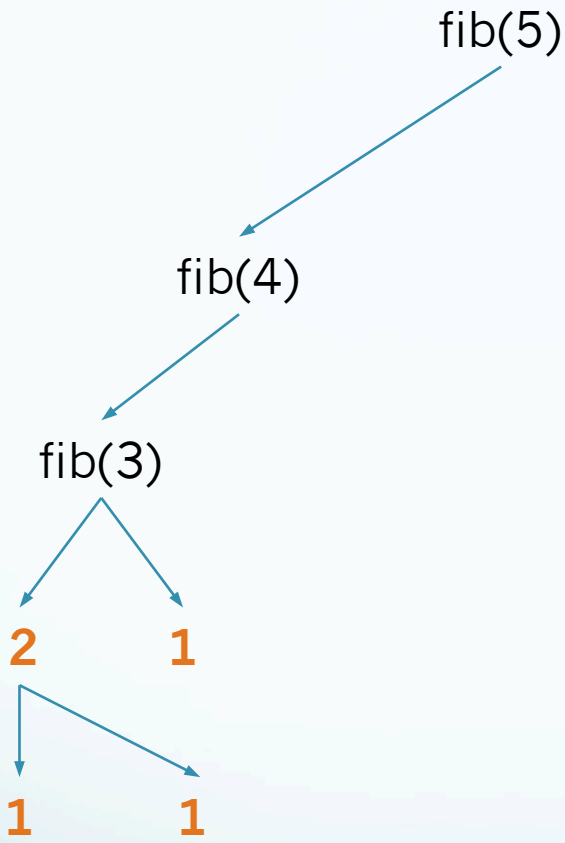



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

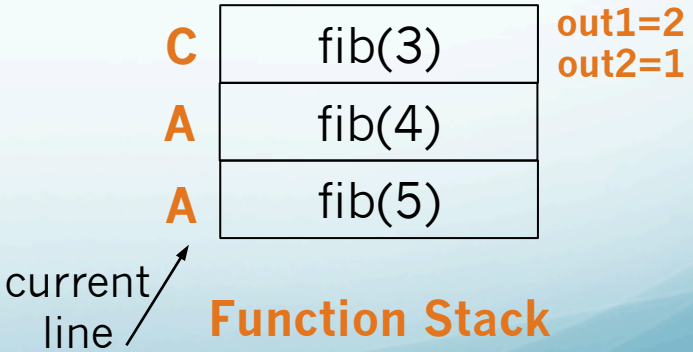
Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`



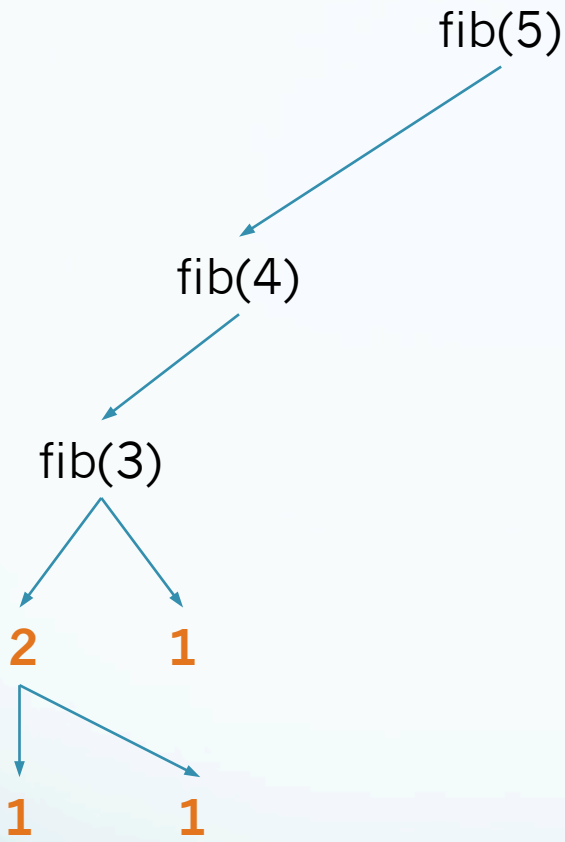
Fibonacci Function Stack



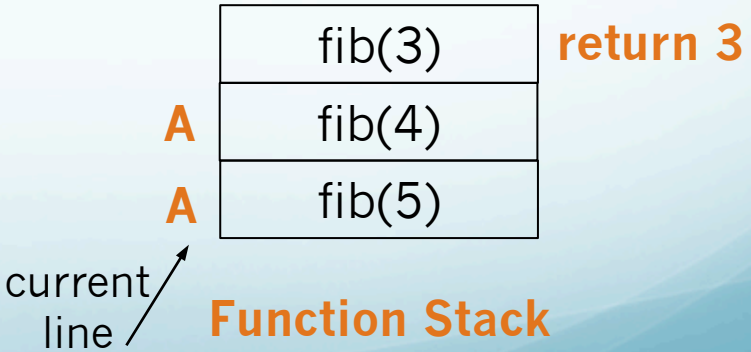

```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
Line A  var out1 = fib(n-1);
Line B  var out2 = fib(n-2);
Line C  return out1+out2;
}
```



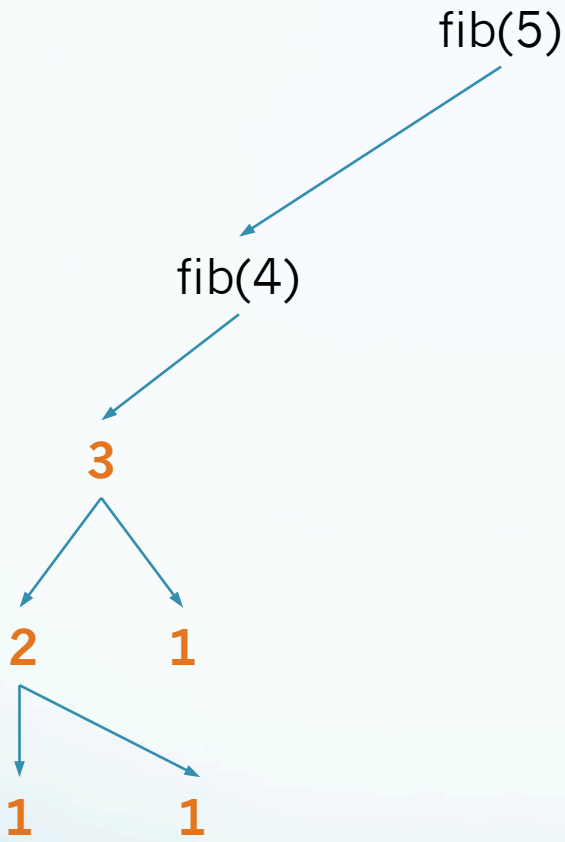
Fibonacci Function Stack




```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
Line A  var out1 = fib(n-1);
Line B  var out2 = fib(n-2);
Line C  return out1+out2;
}
```

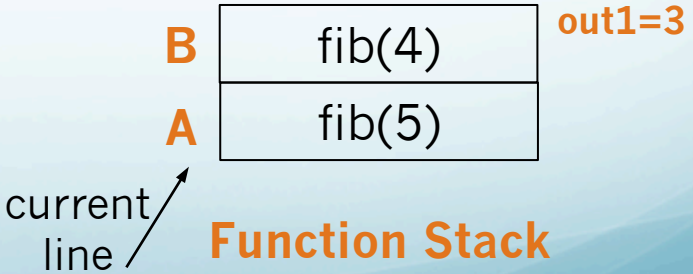


Fibonacci Function Stack

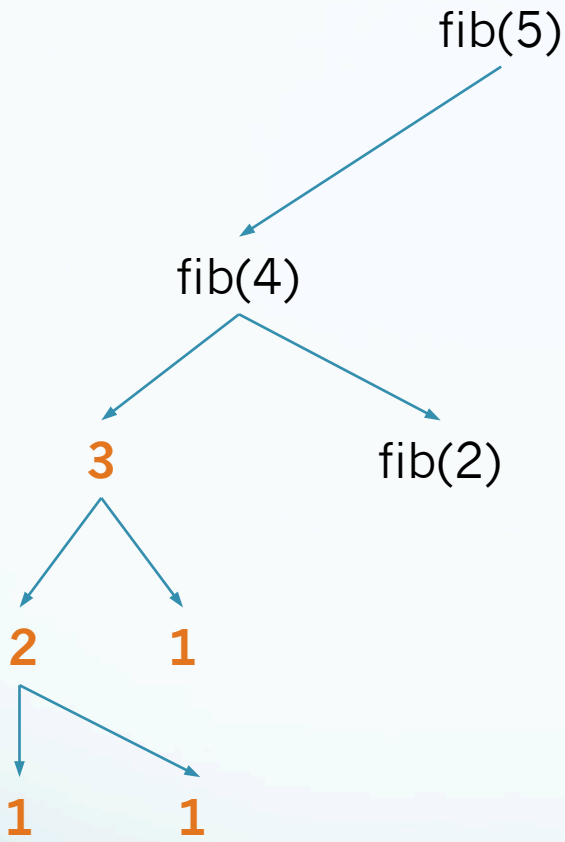



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

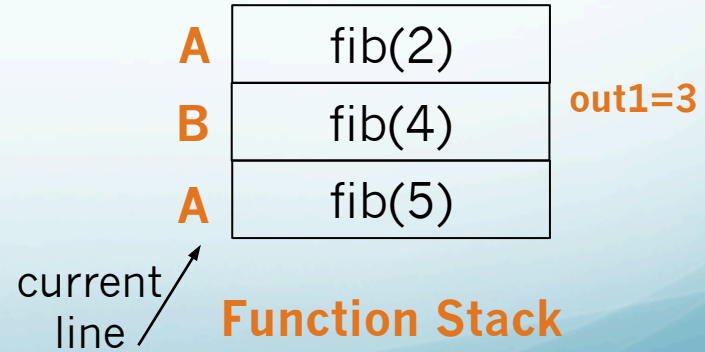


Fibonacci Function Stack

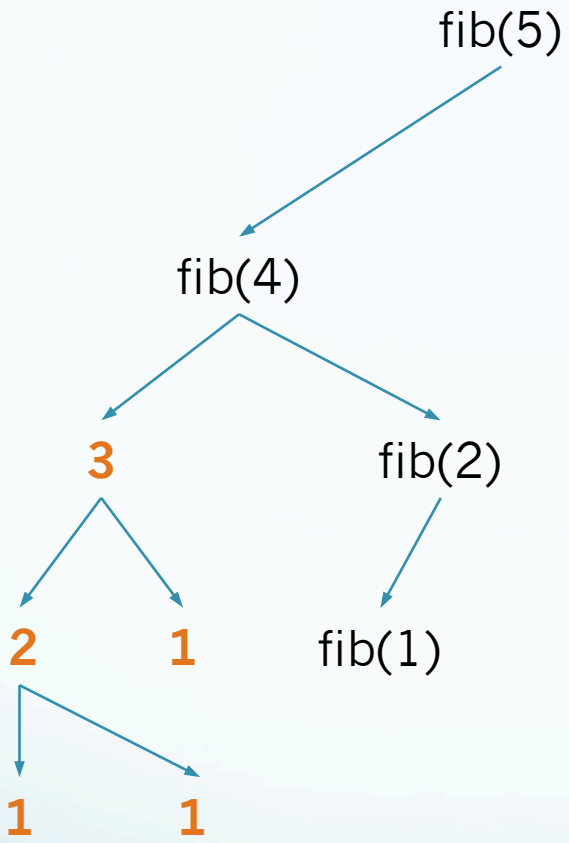



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

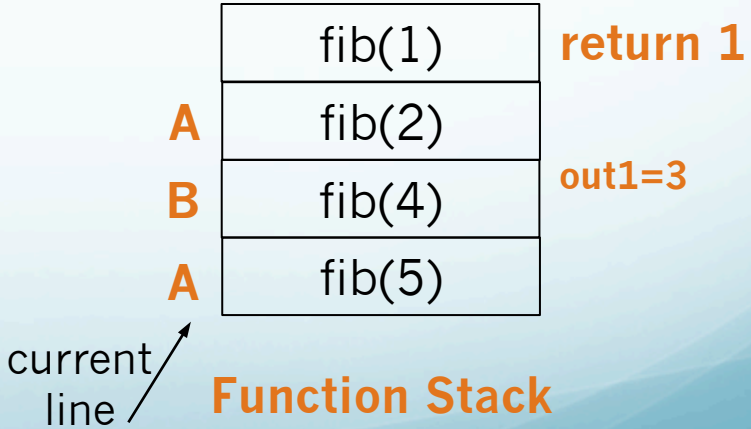


Fibonacci Function Stack

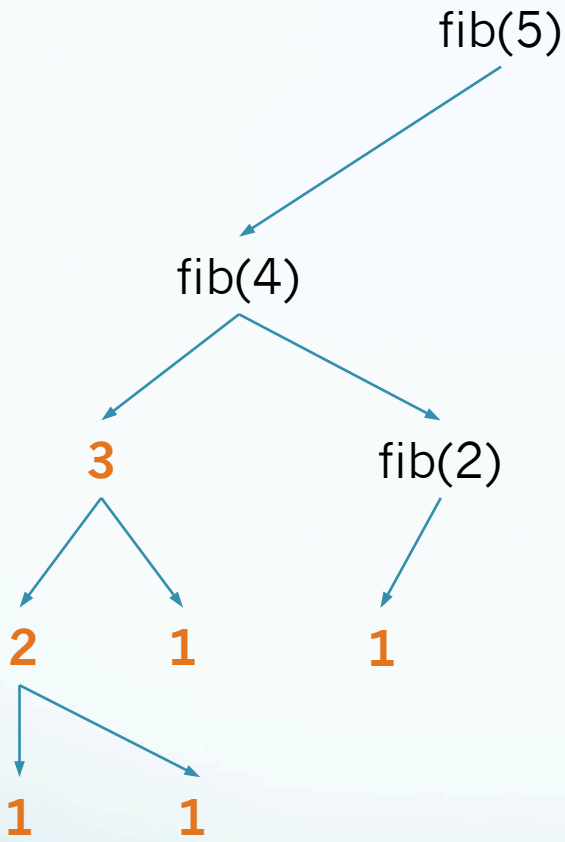



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

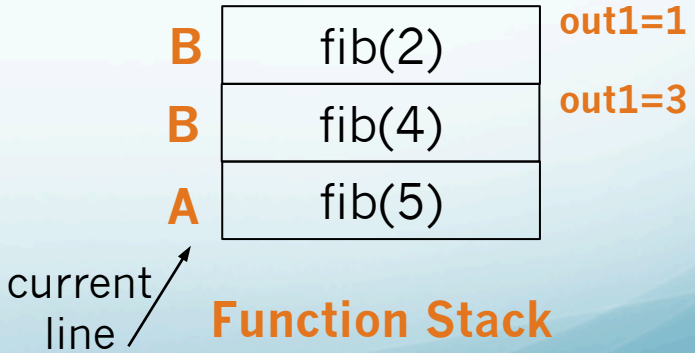
Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`



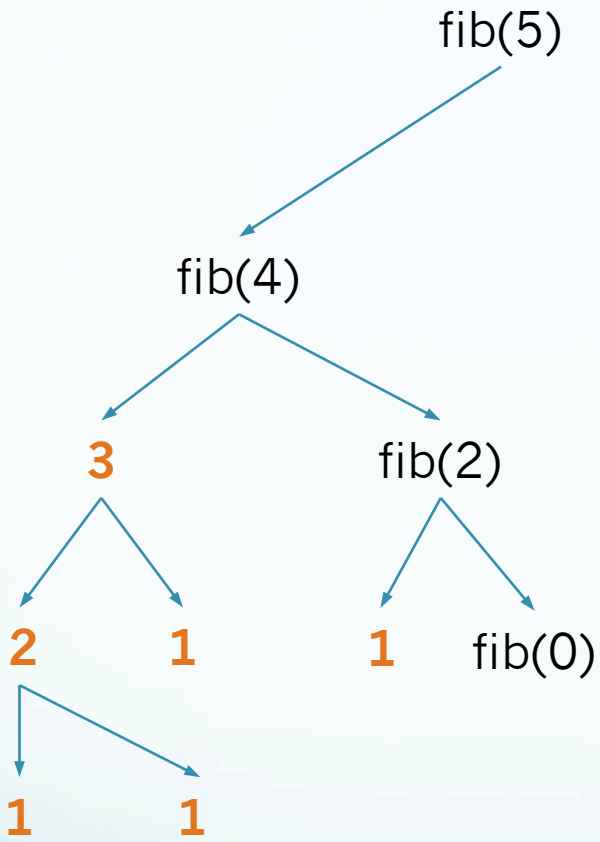
Fibonacci Function Stack



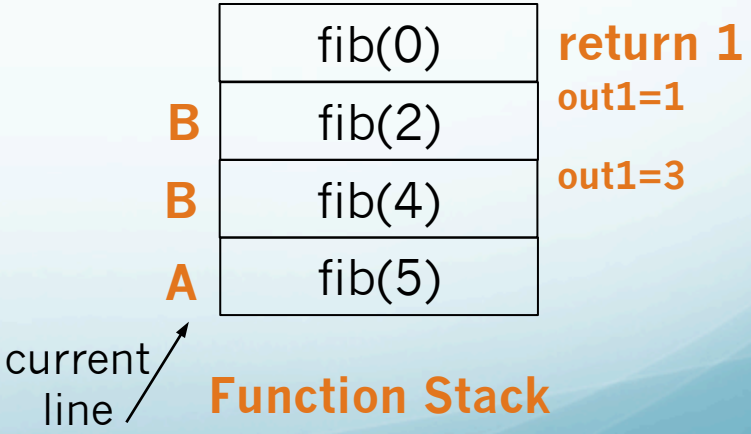

```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
Line A ---> var out1 = fib(n-1);
Line B ---> var out2 = fib(n-2);
Line C ---> return out1+out2;
}
```



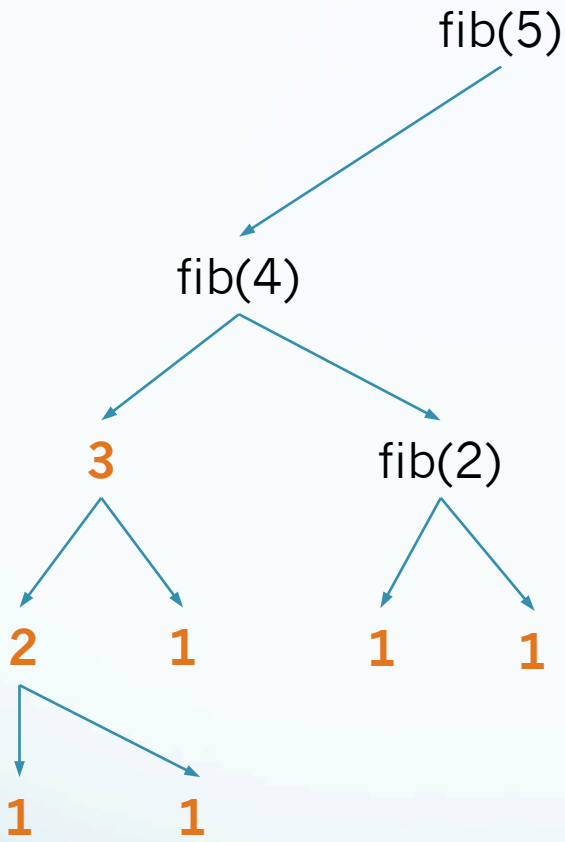
Fibonacci Function Stack




```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
Line A ---> var out1 = fib(n-1);
Line B ---> var out2 = fib(n-2);
Line C ---> return out1+out2;
}
```

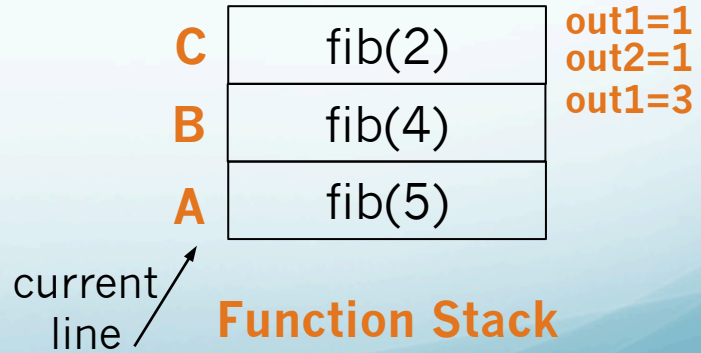


Fibonacci Function Stack

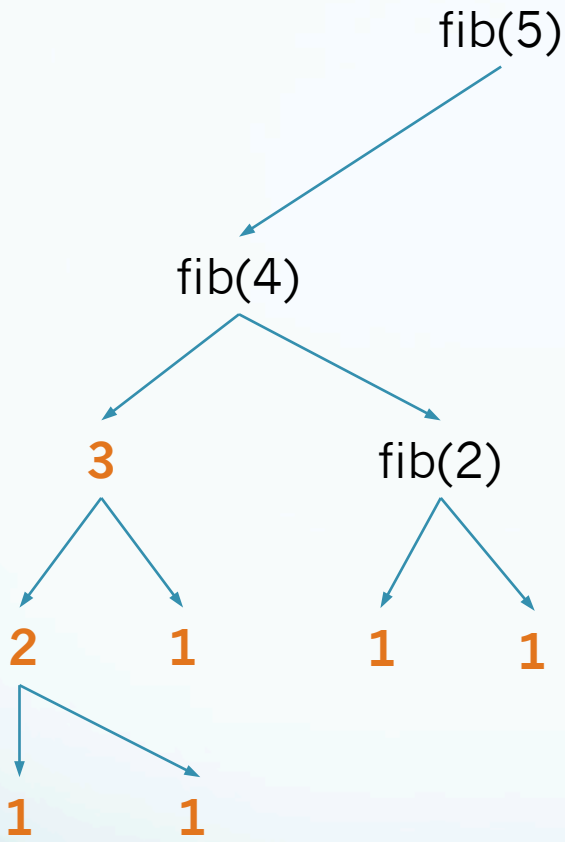



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

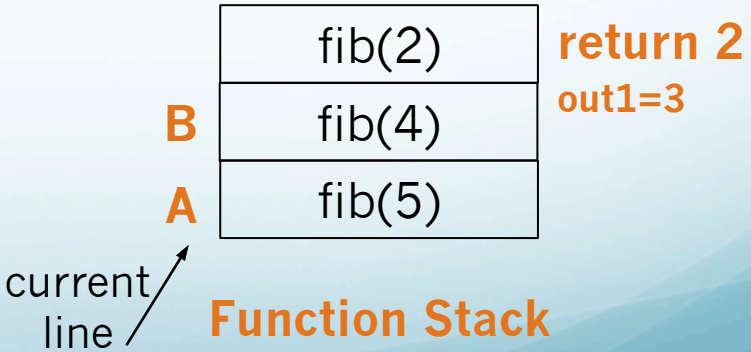


Fibonacci Function Stack

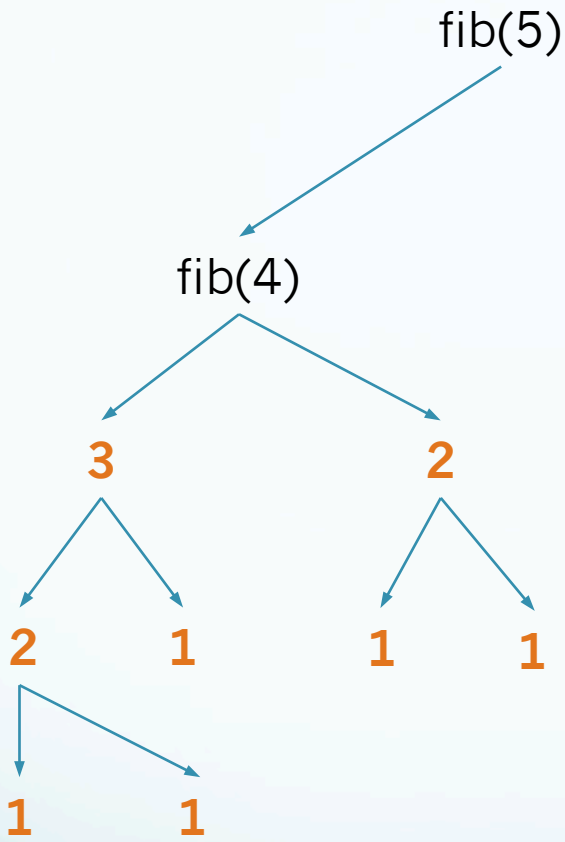



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

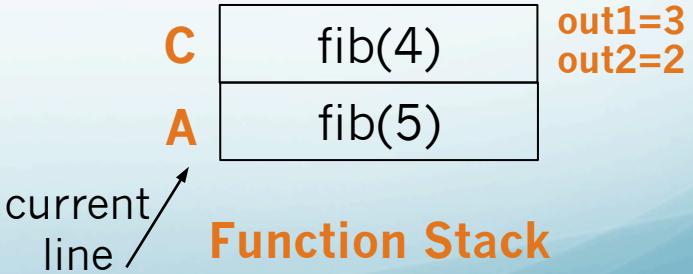
Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`



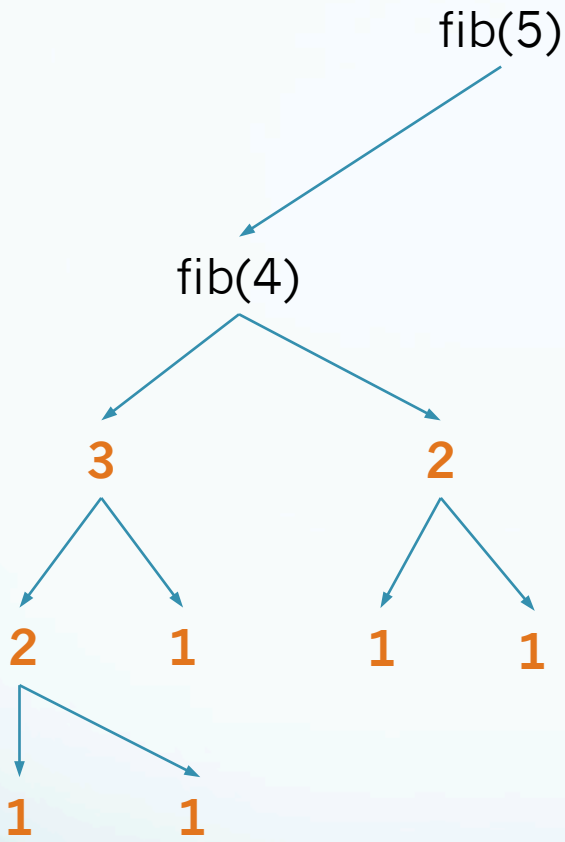
Fibonacci Function Stack



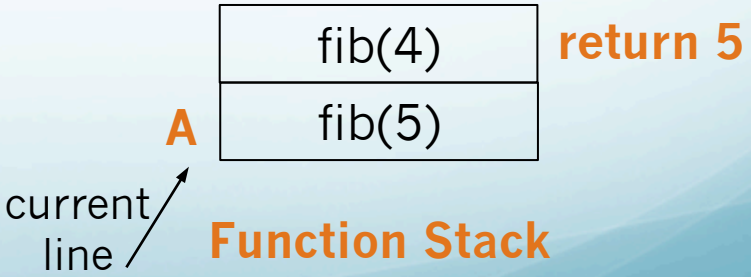

```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
Line A → var out1 = fib(n-1);
Line B → var out2 = fib(n-2);
Line C → return out1+out2;
}
```



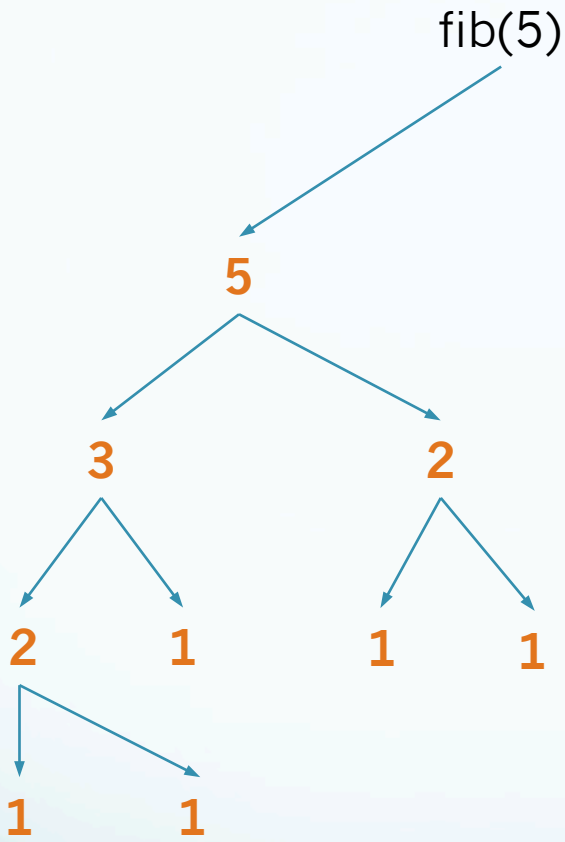
Fibonacci Function Stack



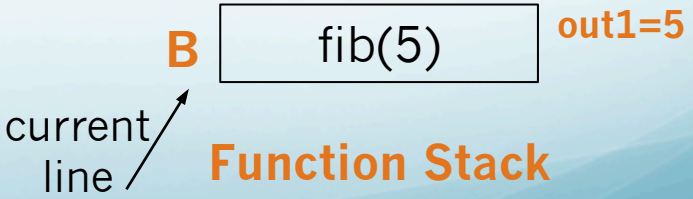

```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
Line A ---> var out1 = fib(n-1);
Line B ---> var out2 = fib(n-2);
Line C ---> return out1+out2;
}
```



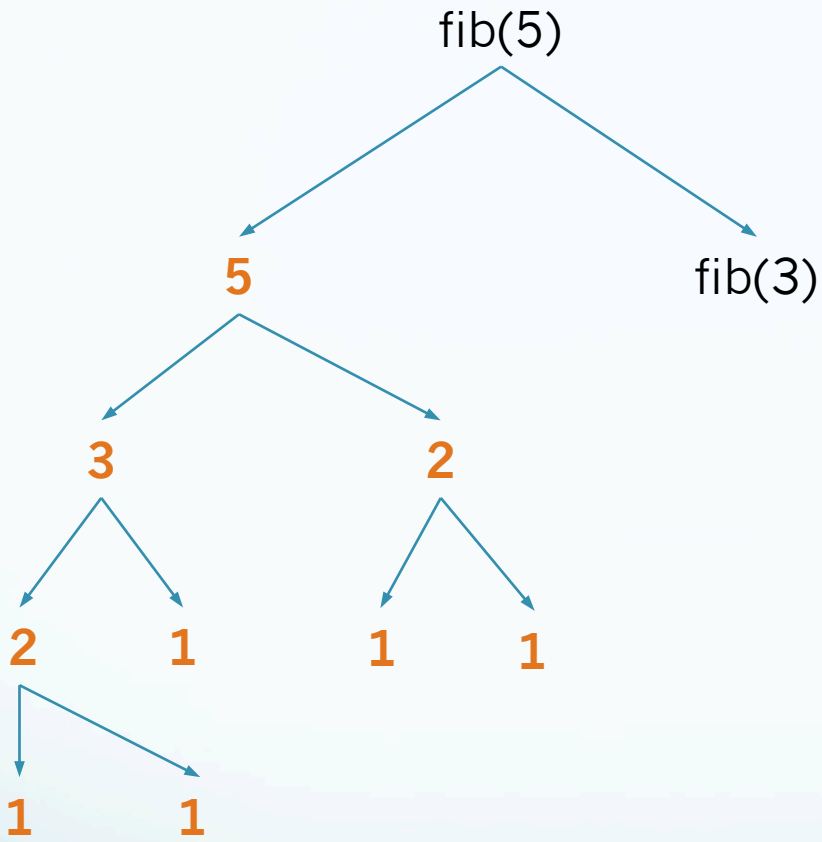
Fibonacci Function Stack




```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
Line A  var out1 = fib(n-1);
Line B  var out2 = fib(n-2);
Line C  return out1+out2;
}
```

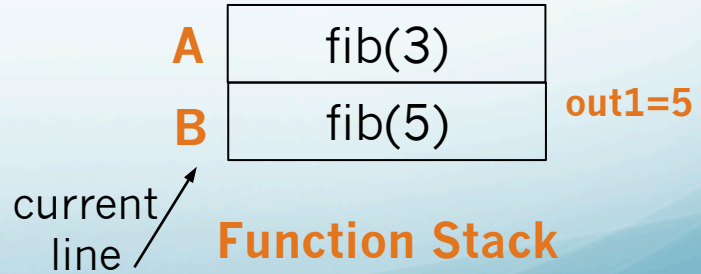


Fibonacci Function Stack

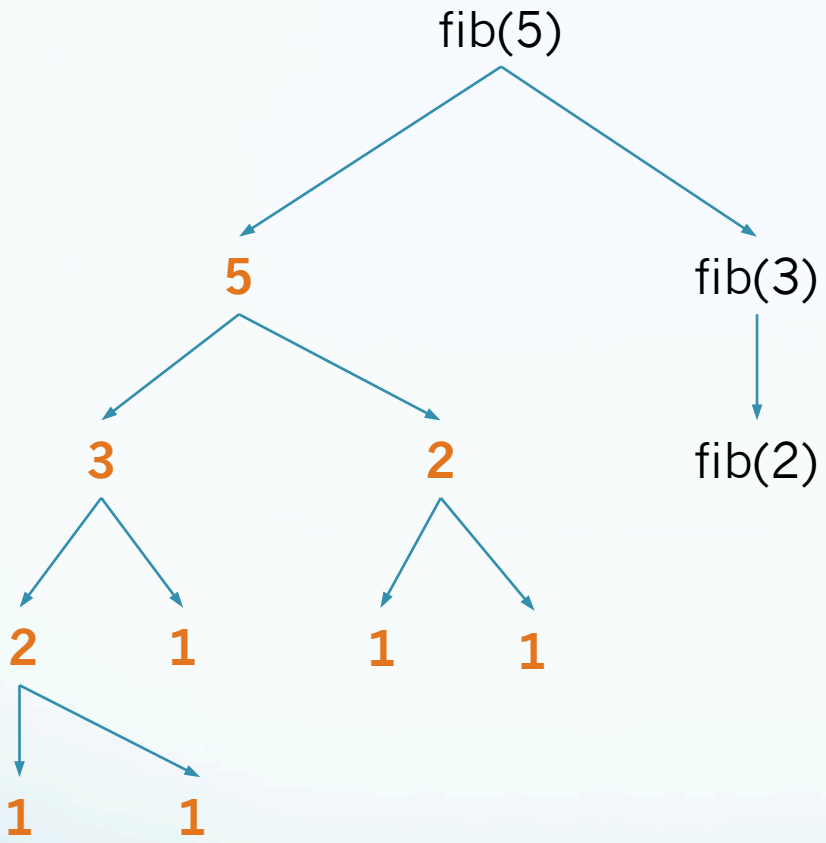



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

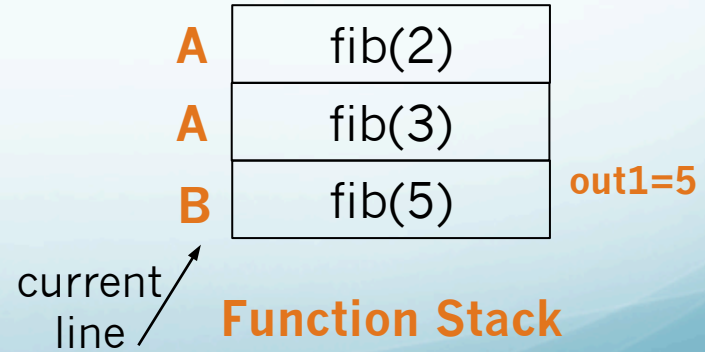


Fibonacci Function Stack

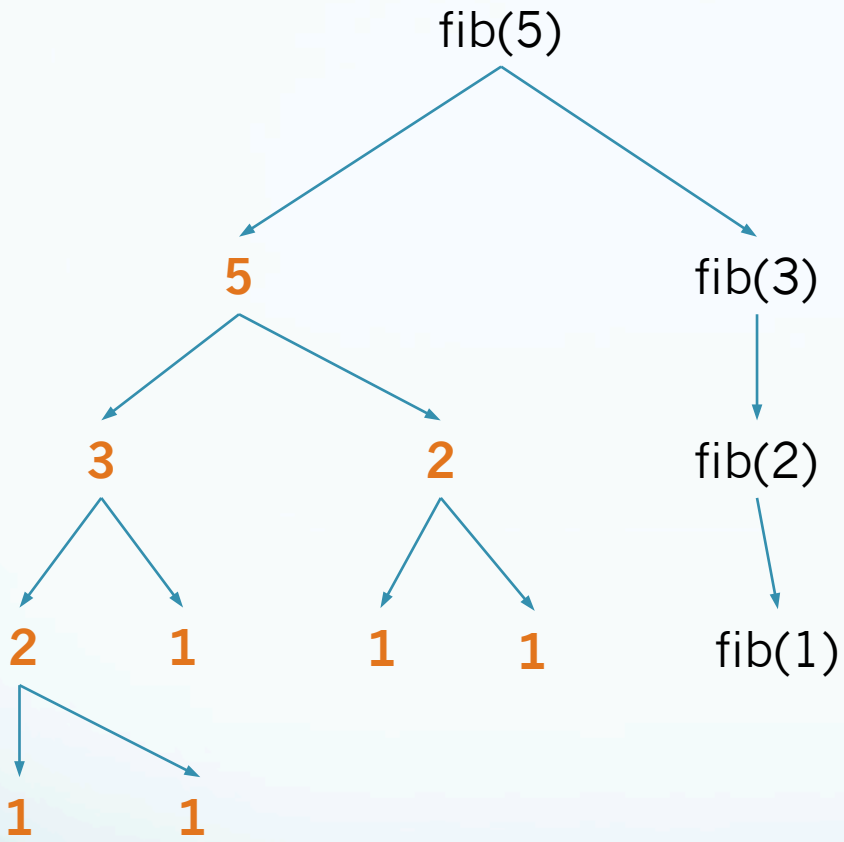



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → var out1 = fib(n-1);
Line B → var out2 = fib(n-2);
Line C → return out1+out2;

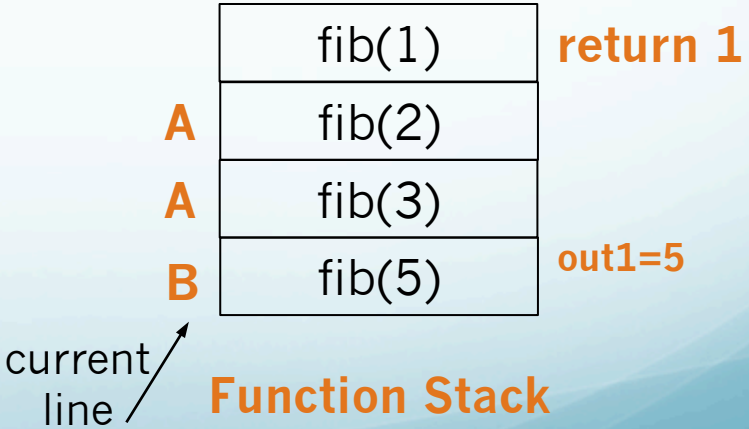


Fibonacci Function Stack

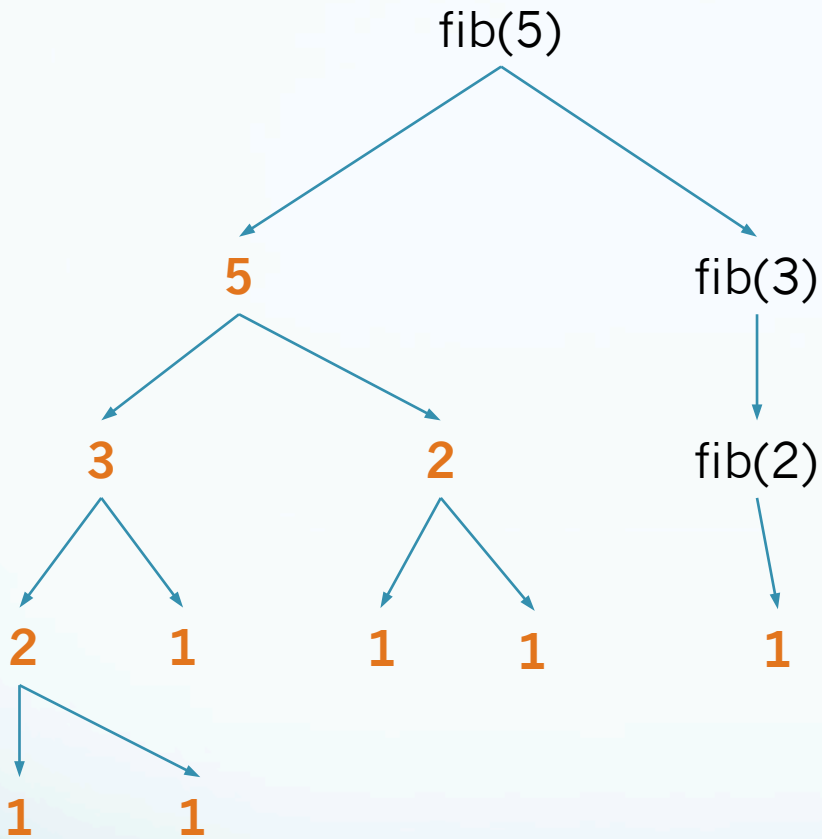



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

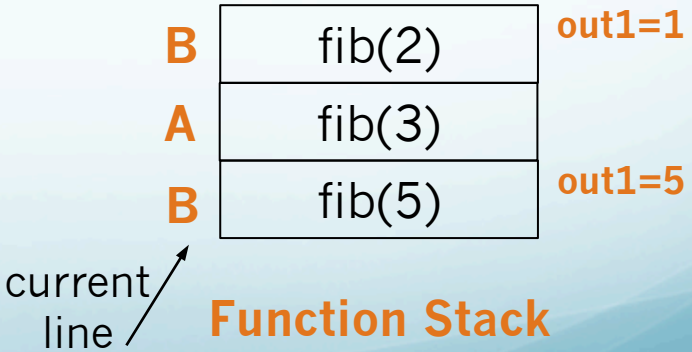


Fibonacci Function Stack

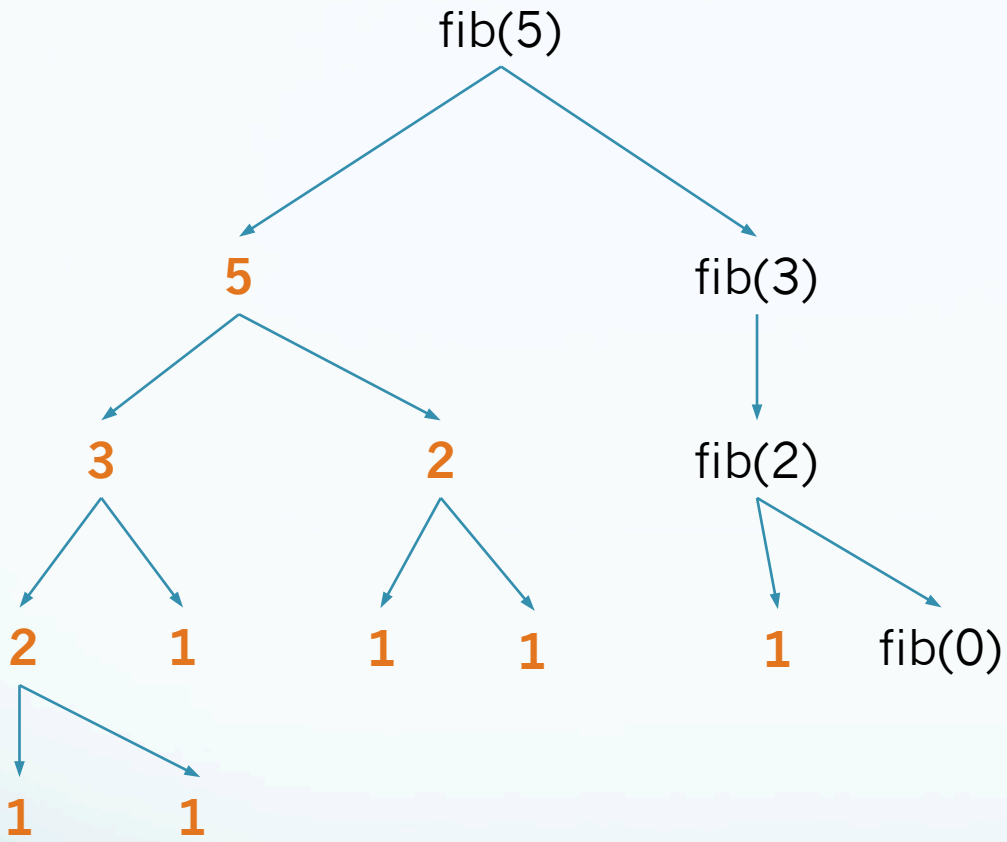



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

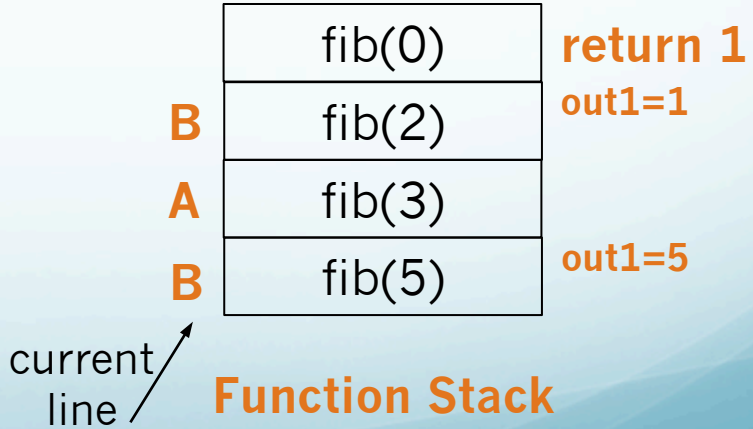


Fibonacci Function Stack

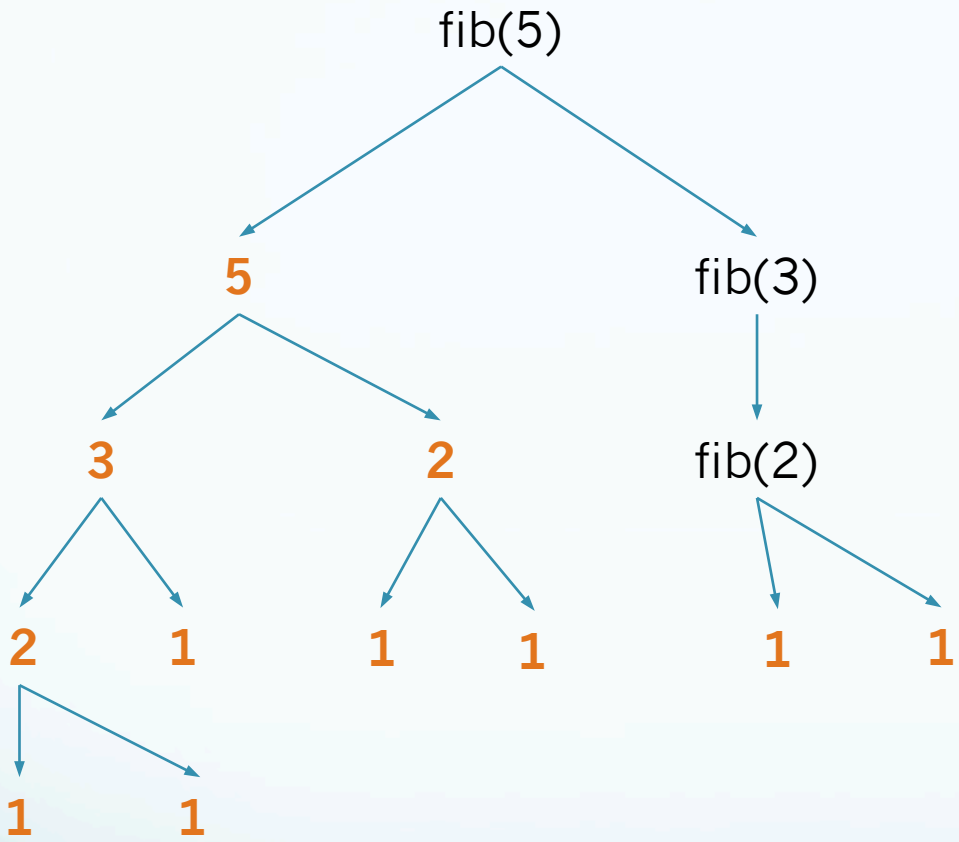



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

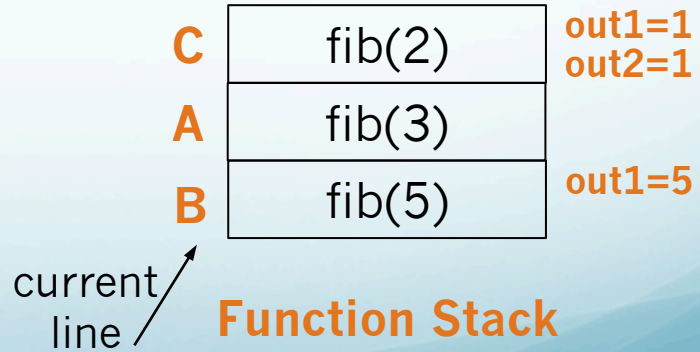


Fibonacci Function Stack

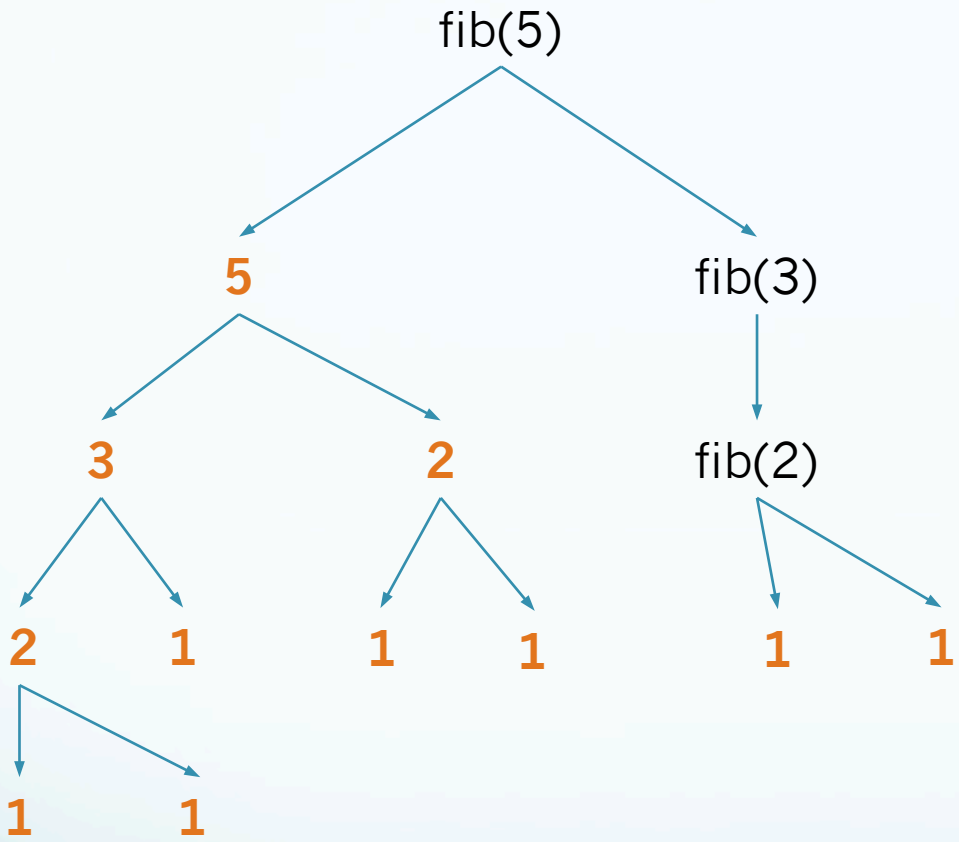



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

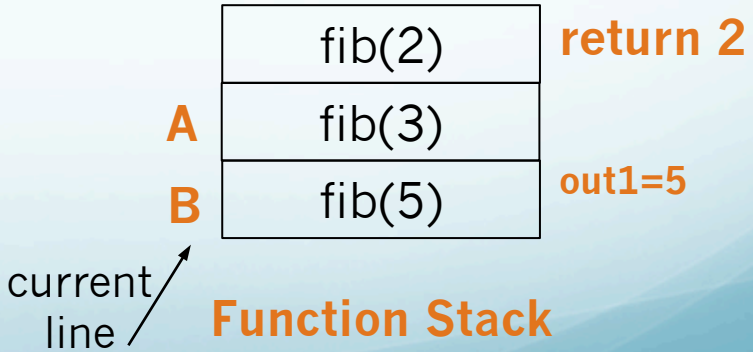


Fibonacci Function Stack

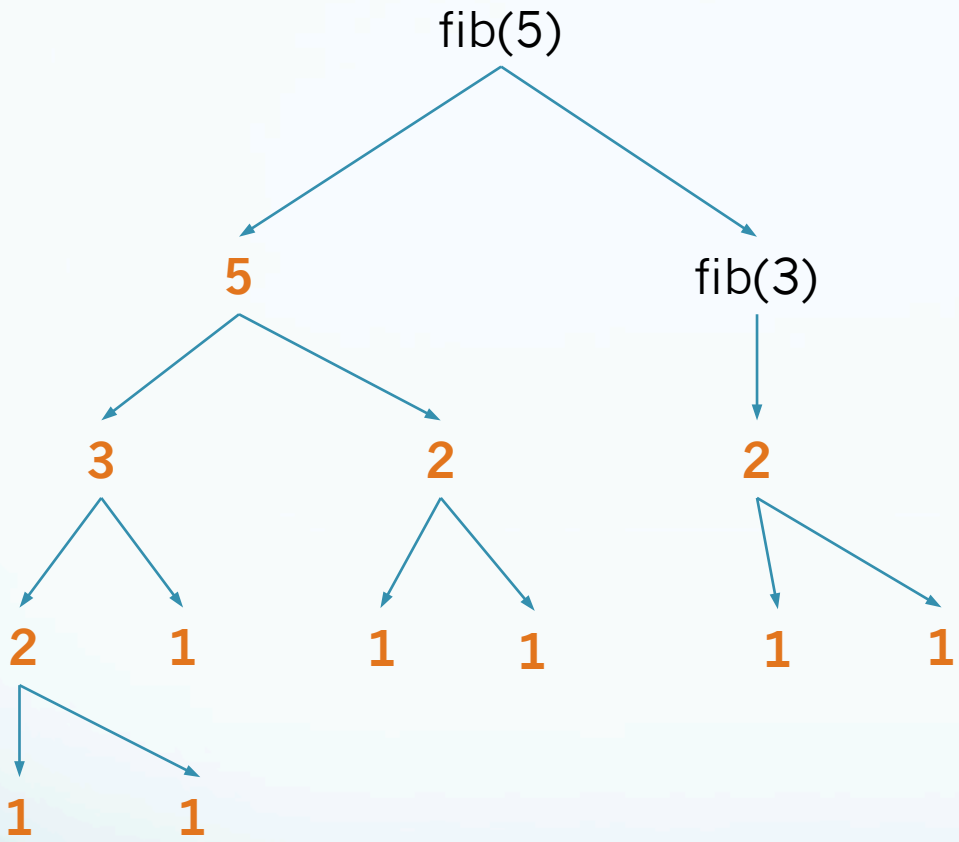



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

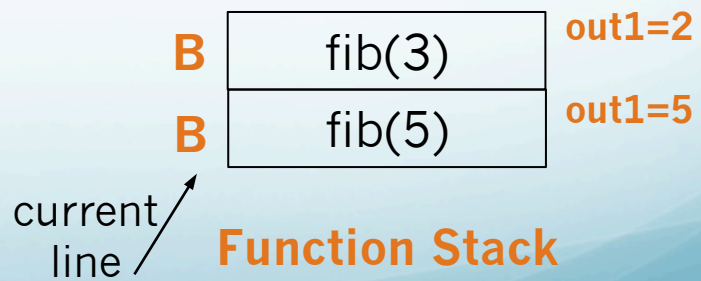


Fibonacci Function Stack

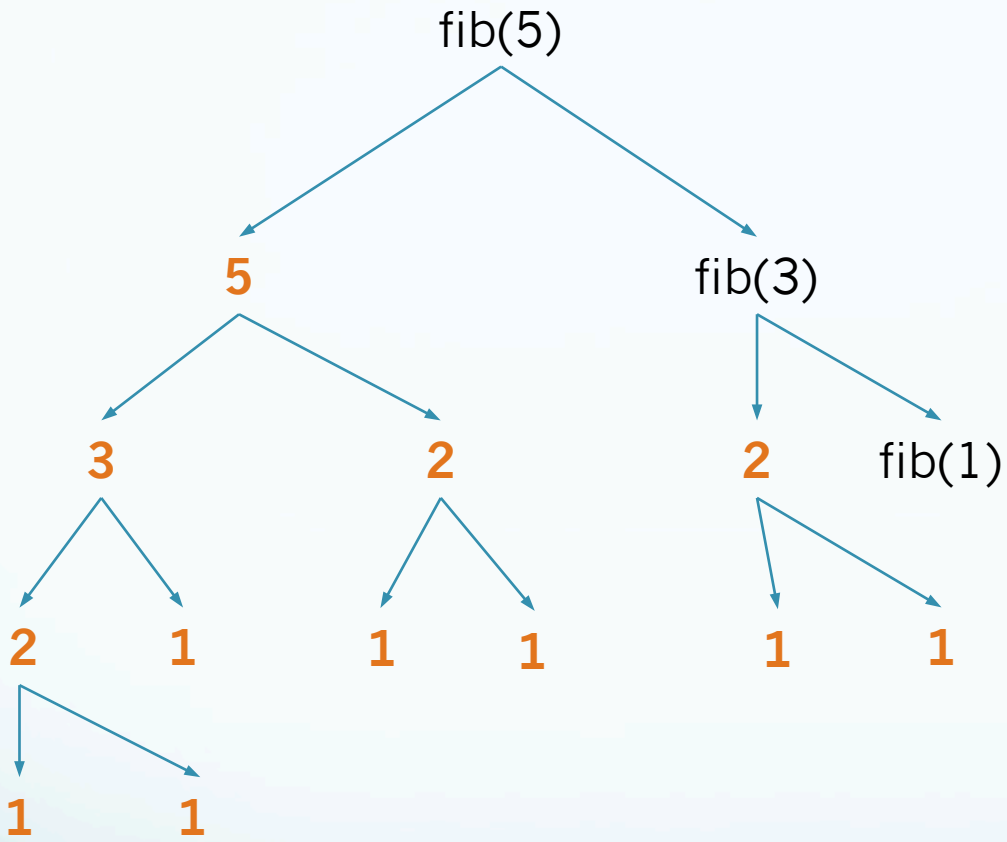



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

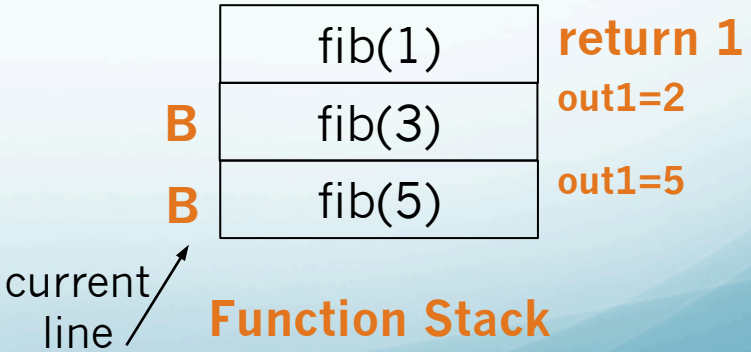


Fibonacci Function Stack

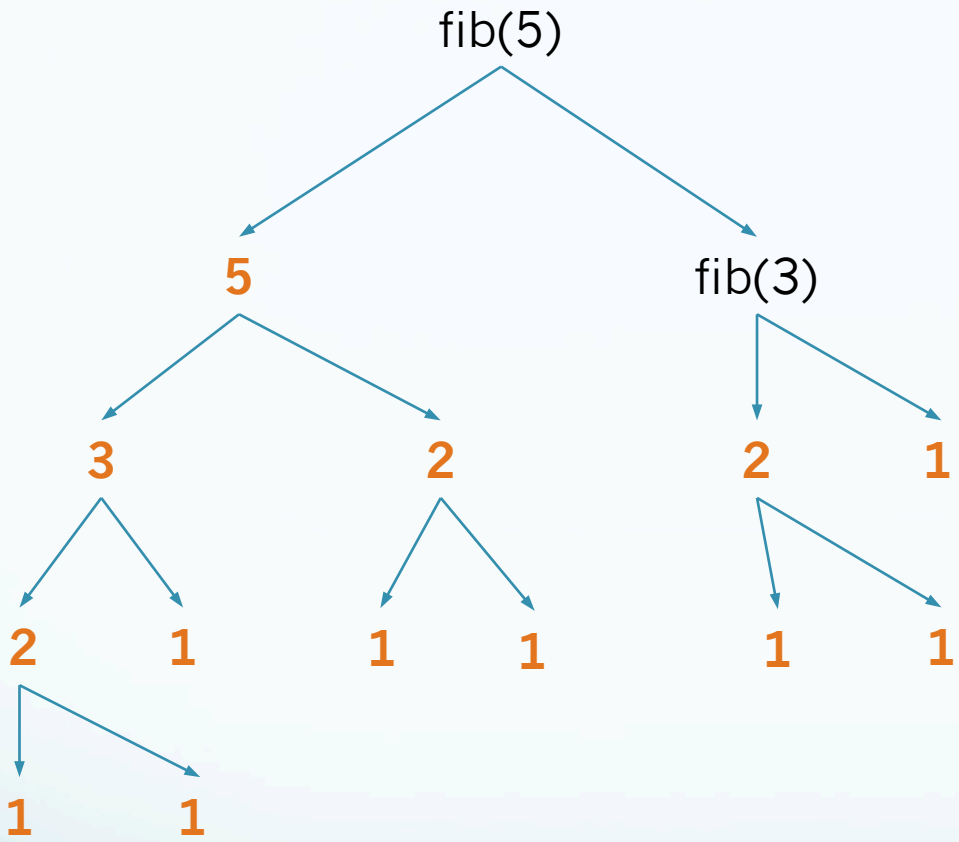



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

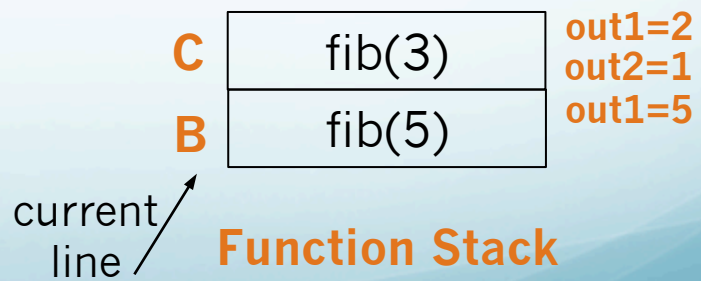


Fibonacci Function Stack

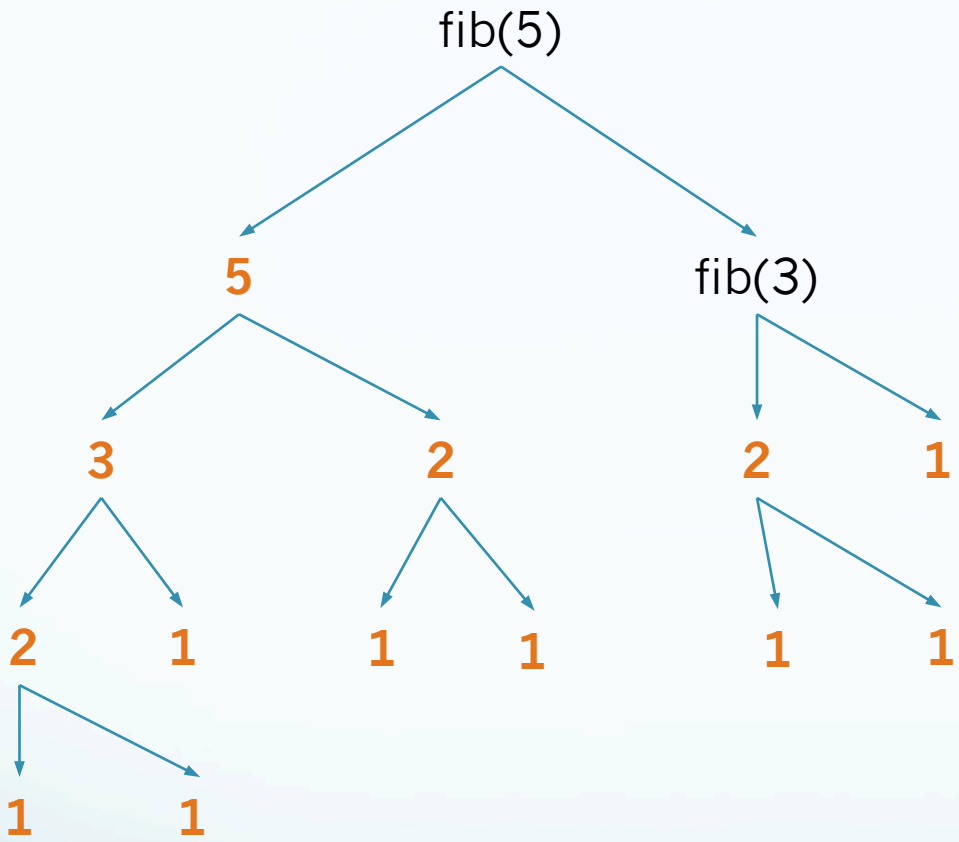



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

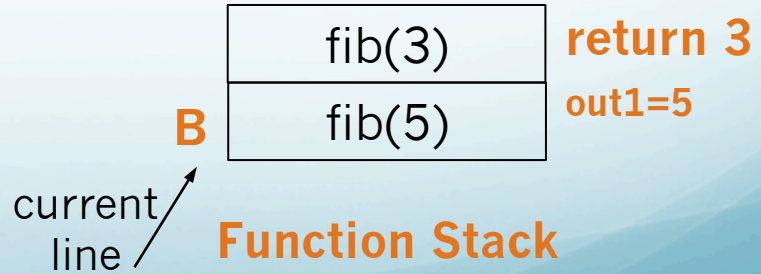


Fibonacci Function Stack

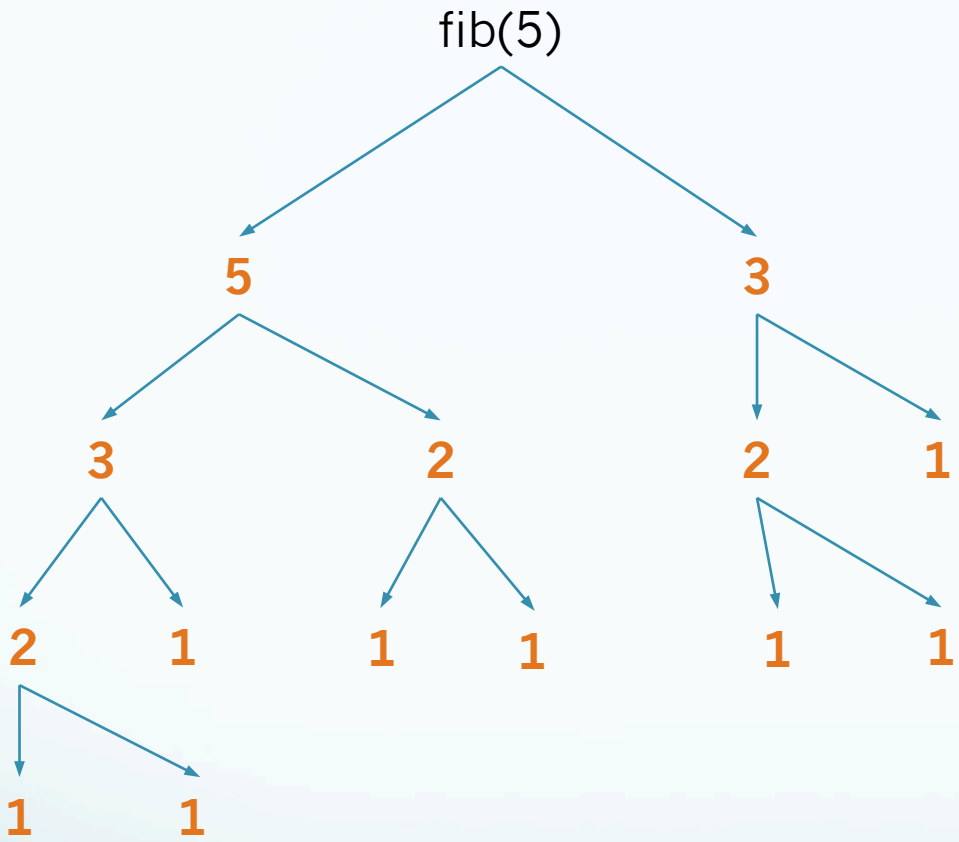



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

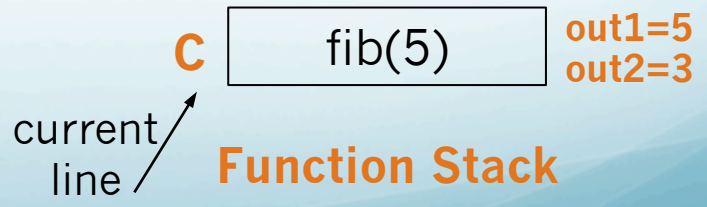


Fibonacci Function Stack

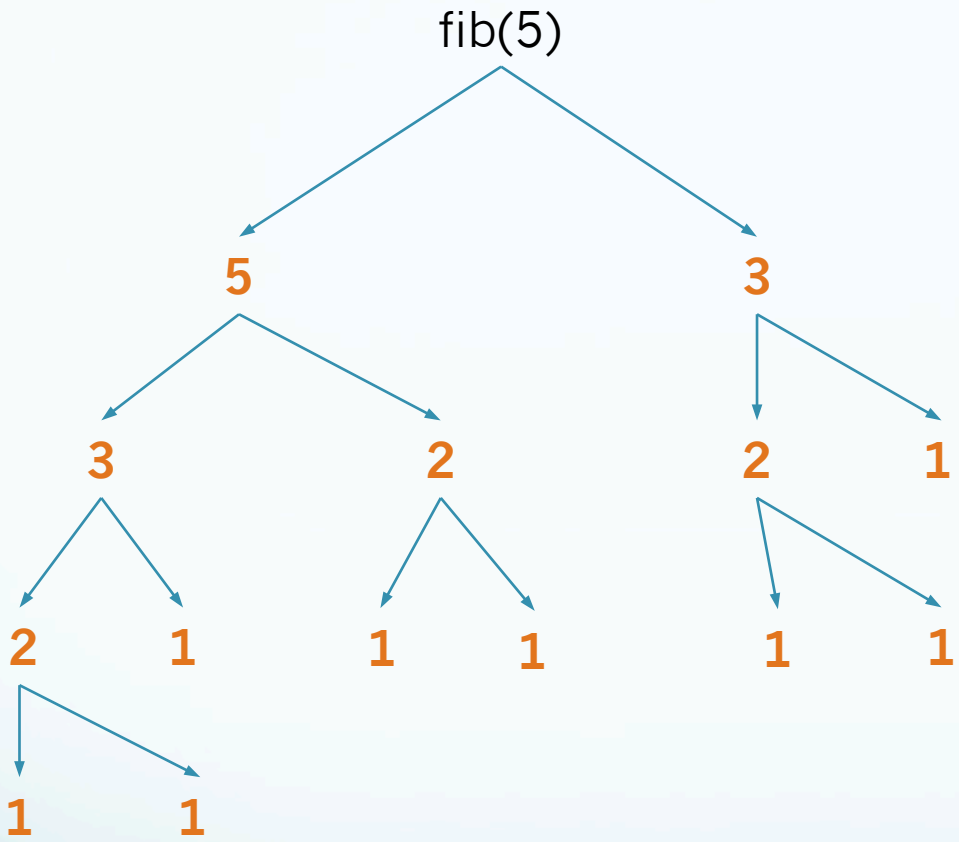



```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`



Fibonacci Function Stack




```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

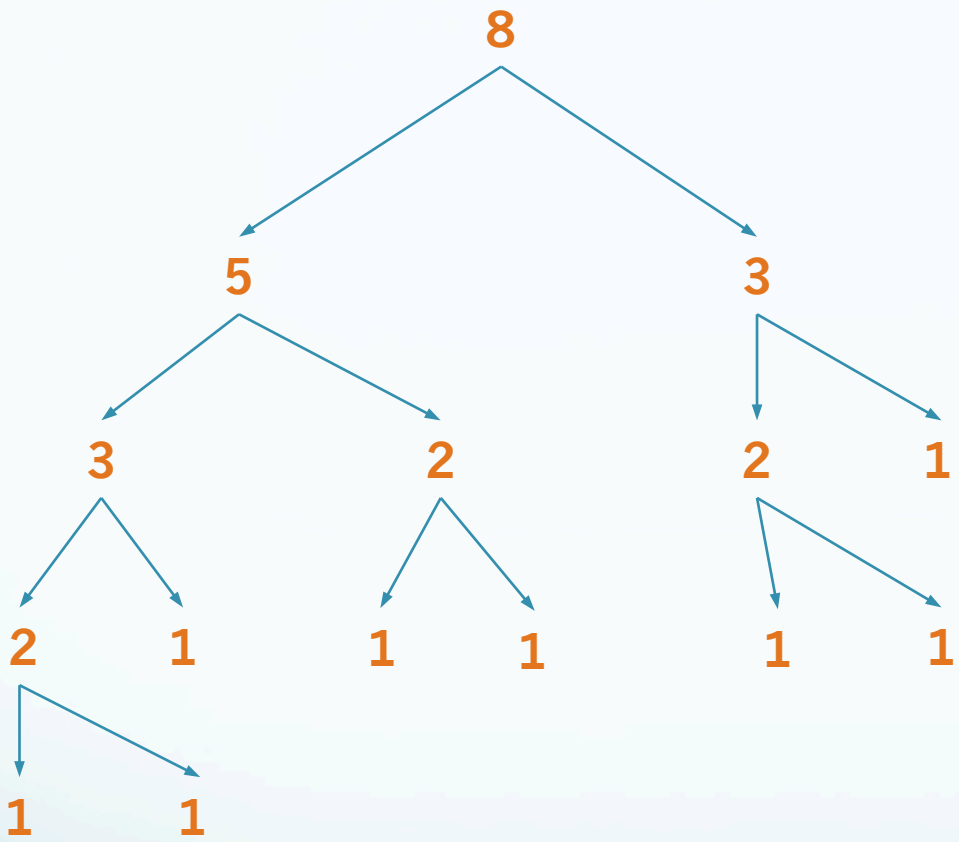
Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

current line

fib(5) **return 8**

Function Stack

Fibonacci Function Stack



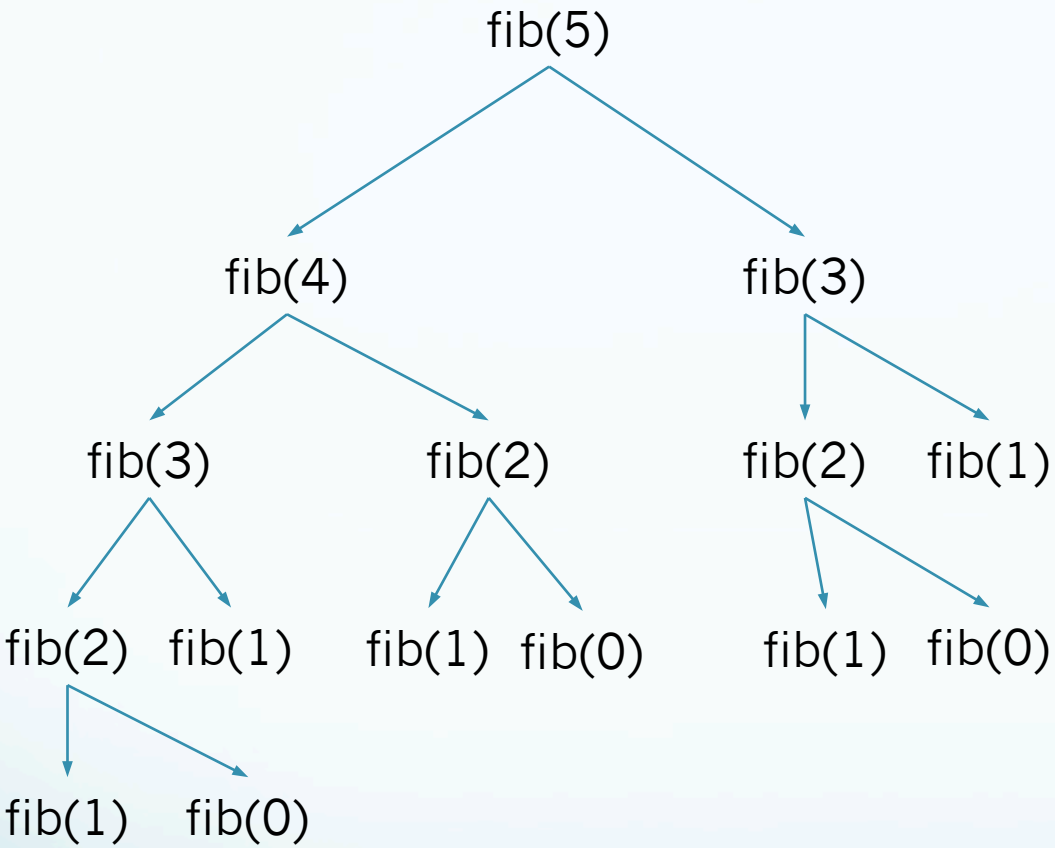

```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

empty!

Function Stack

Fibonacci Tree with Function Calls




```
function fib(n) {
    if (n==0 || n==1) {
        return 1;
    }
    var out1 = fib(n-1);
    var out2 = fib(n-2);
    return out1+out2;
}
```

Line A → `var out1 = fib(n-1);`
Line B → `var out2 = fib(n-2);`
Line C → `return out1+out2;`

Fill order activity, Sweep Fill