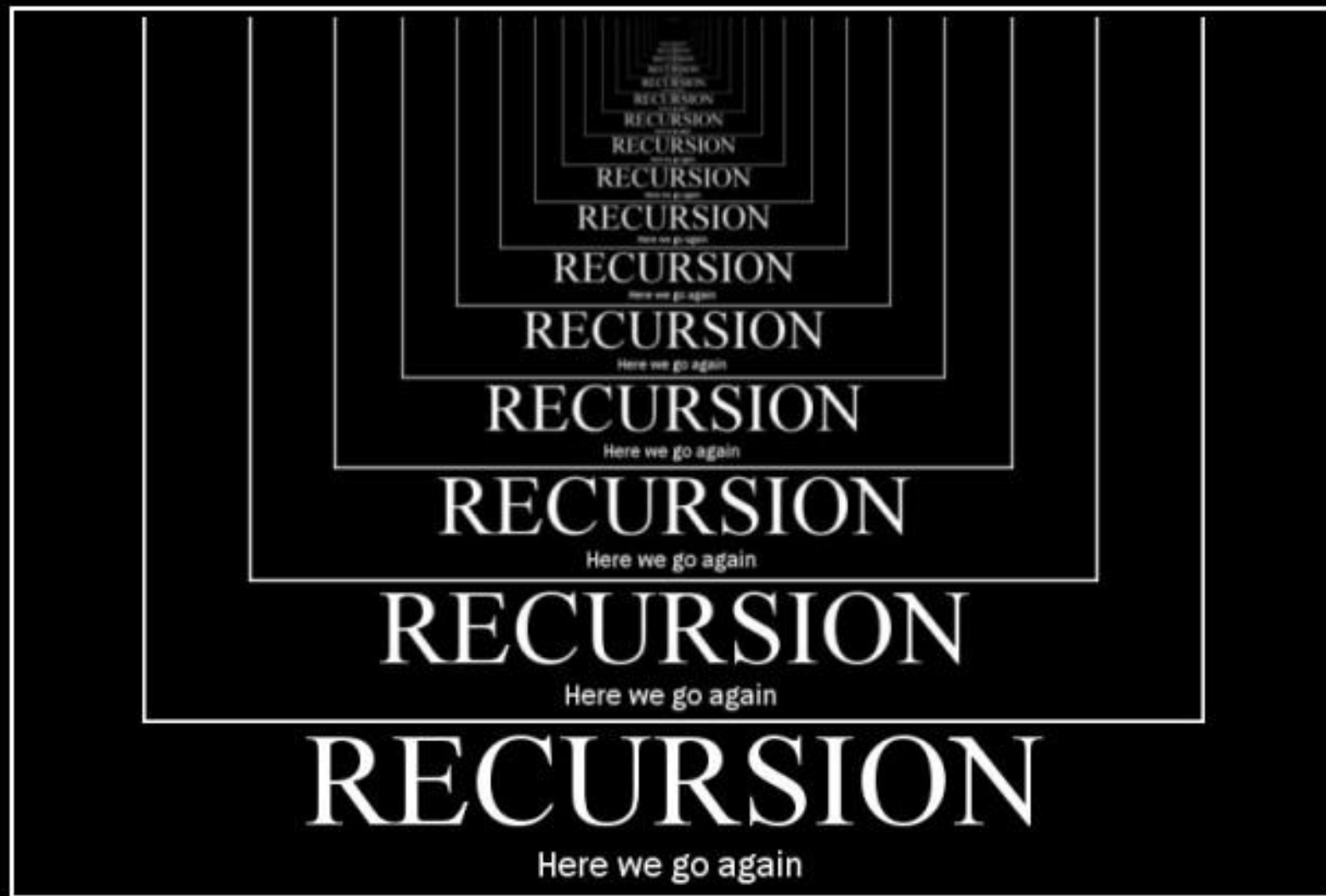
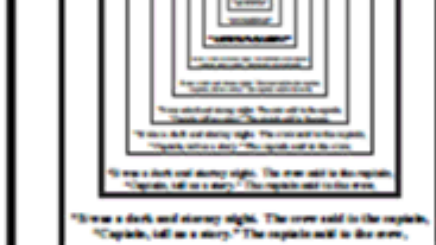


Recursion Calls



RECURSION

Here we go again



"It was a dark and stormy night. The crew said to the captain,
"Captain, tell us a story." The captain said to the crew,

"It was a dark and stormy night. The crew said to the captain,
"Captain, tell us a story." The captain said to the crew,

"It was a dark and stormy night. The crew said to the captain,
"Captain, tell us a story." The captain said to the crew,

"It was a dark and stormy night. The crew said to the captain,
"Captain, tell us a story." The captain said to the crew,

"It was a dark and stormy night. The crew said to the captain,
"Captain, tell us a story." The captain said to the crew,

"It was a dark and stormy night. The crew said to the captain,
"Captain, tell us a story." The captain said to the crew,

"It was a dark and stormy night. The crew said to the captain,
"Captain, tell us a story." The captain said to the crew,

"It was a dark and stormy night. The crew said to the captain,
"Captain, tell us a story." The captain said to the crew,







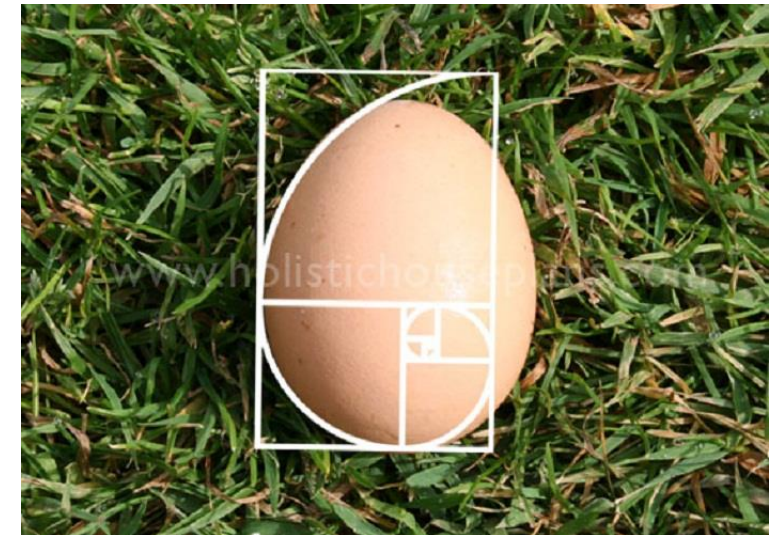
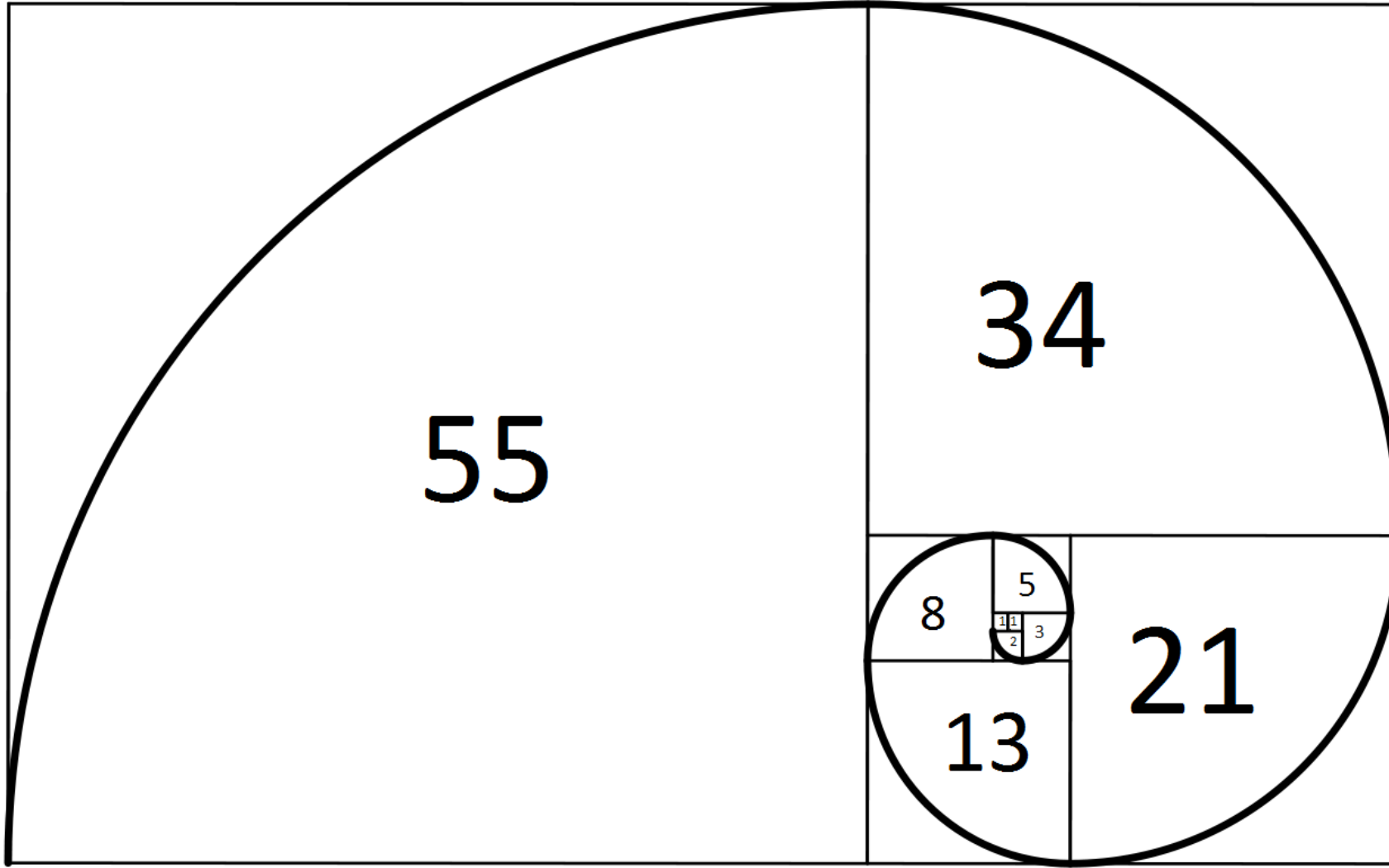


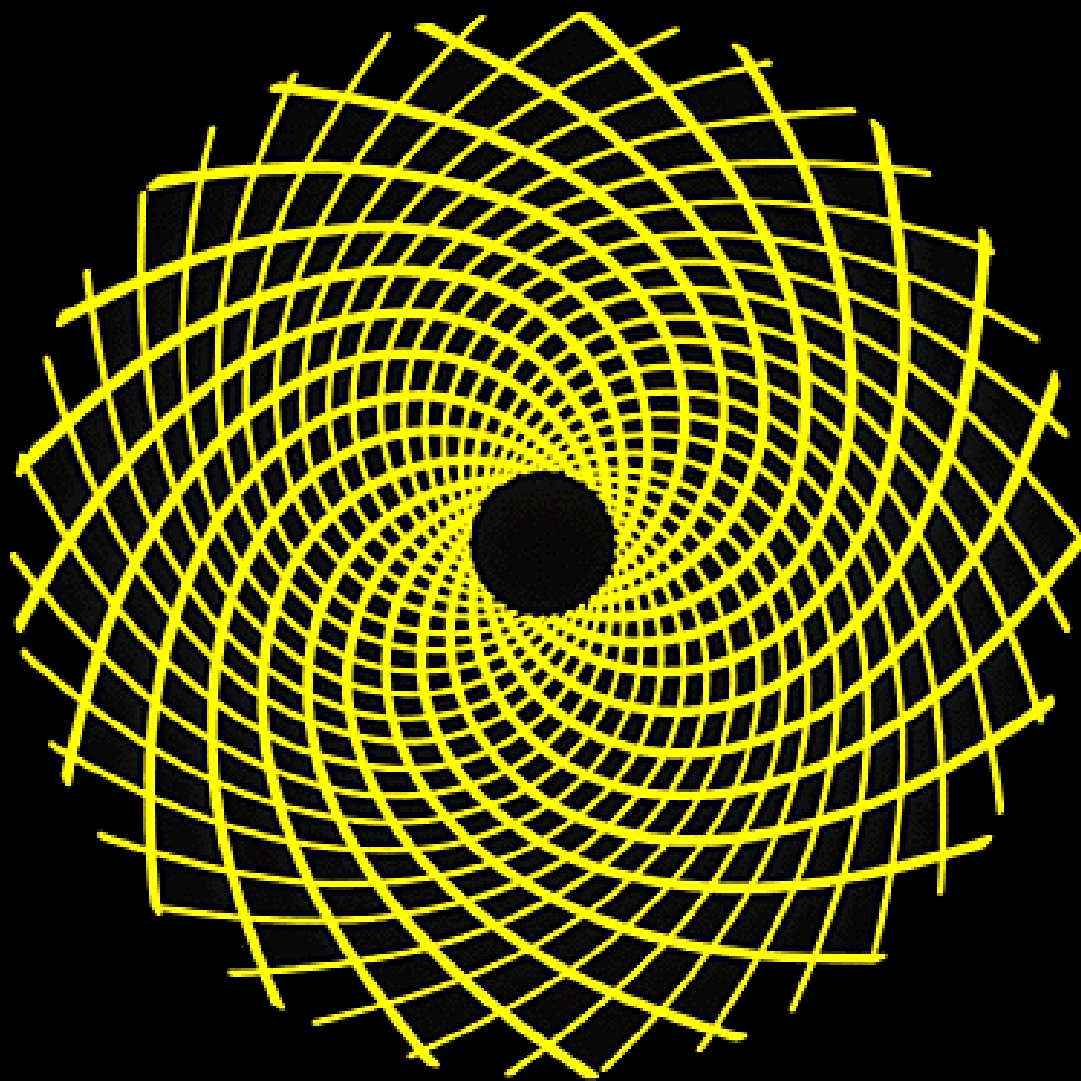
1	2	3	4	5	6	7	8	9	10	11	...
1	1	2	3	5	8	13	21	34	55	89	...

What does fib(6) return?

What does fib(1) return?

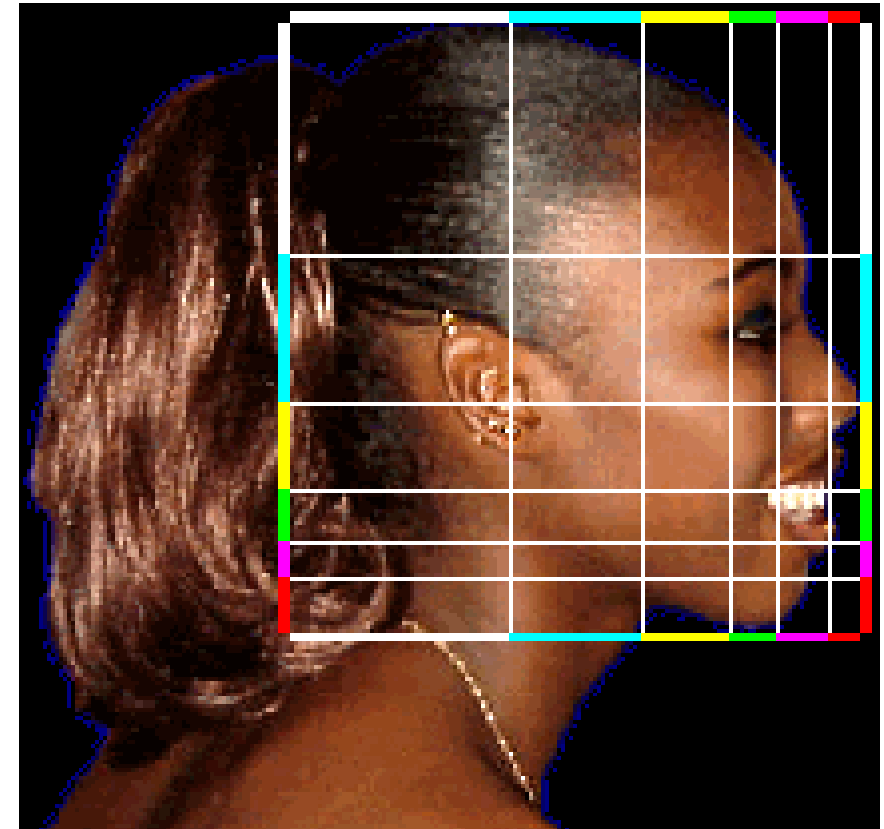
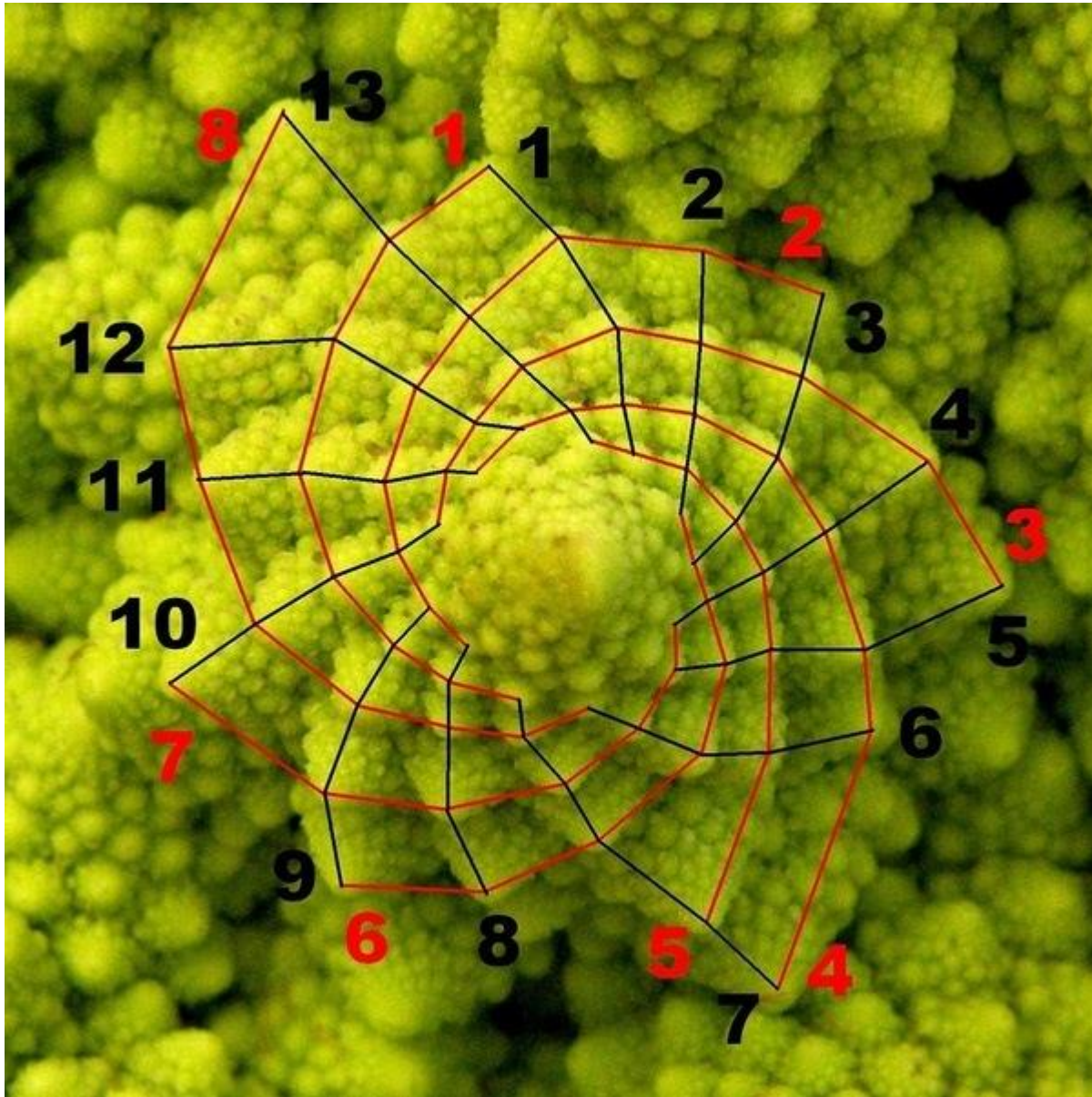
What does fib(9) return?

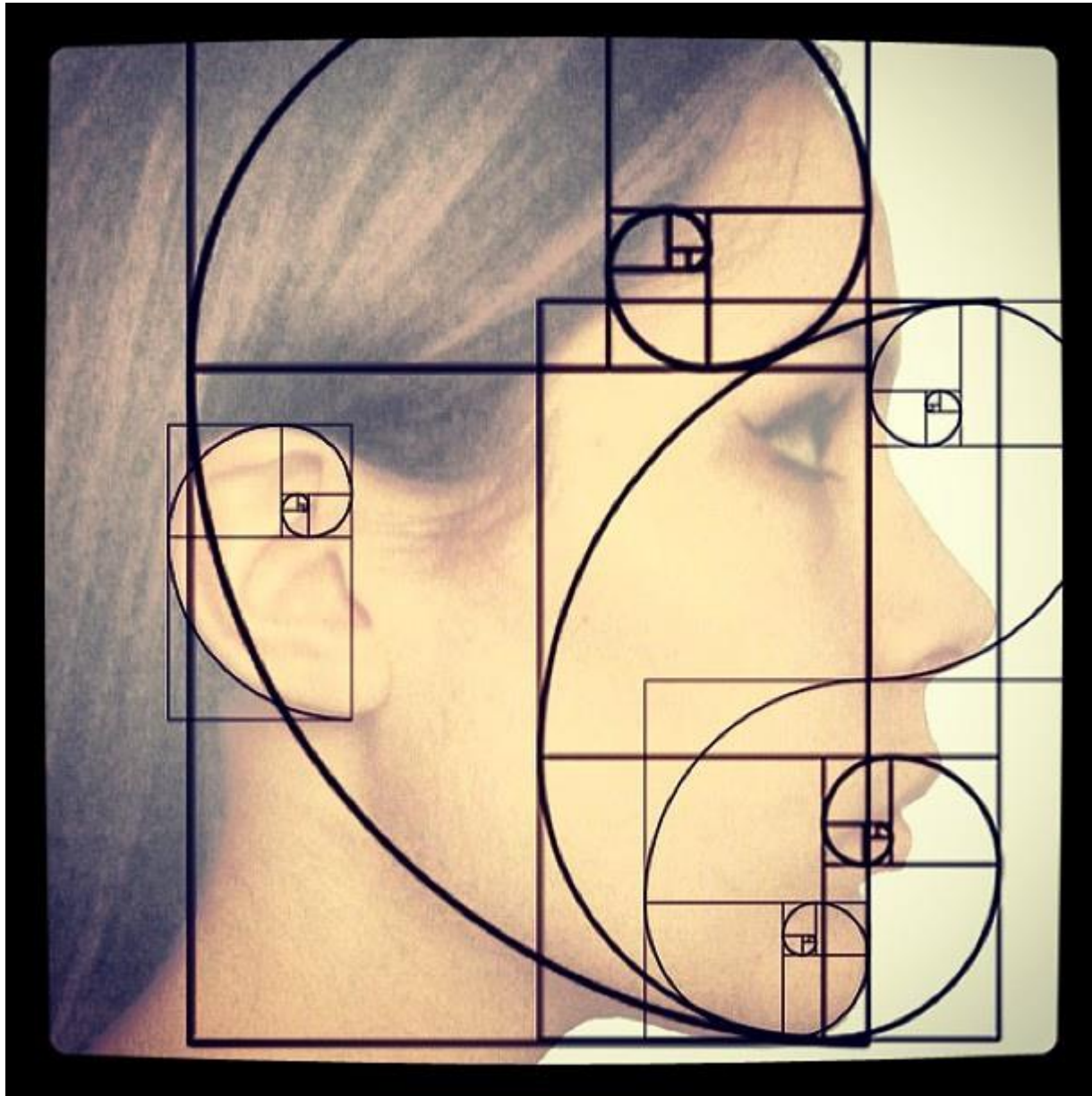




21







54 mm

86 mm

$$54/86 = .6279$$

$$\text{Golden Ratio} = .618$$

If the Fibonacci number is in position 1 or 2,

It is one

Otherwise

It is the sum of two previous Fibonacci numbers

```
System.out.println(fib(5));
```

```
fib(5)
```

1	2	3	4	5	6	7	8	9	10	11	...
1	1	2	3	5	8	13	21	34	55	89	...

```
public int fib (int n) {  
    if (n == 1 || n == 2)  
        return 1;  
    else  
        return fib (n - 1) + fib (n - 2);  
}
```



```
System.out.println(fib(5));
```

fib(5)

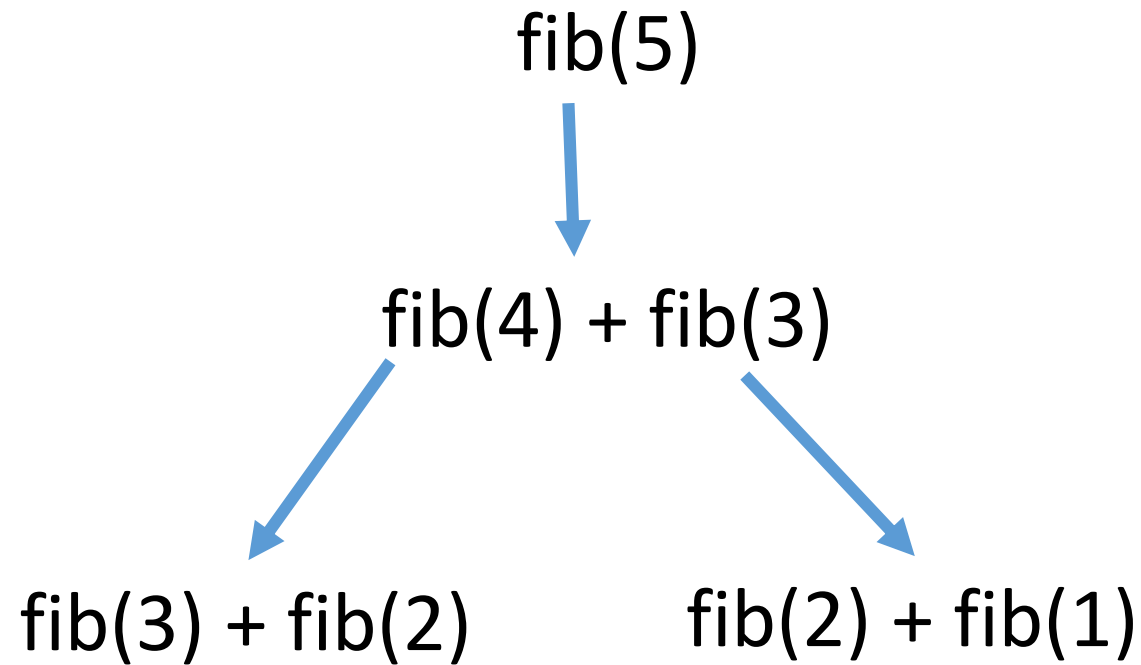
fib(4) + fib(3)

1	2	3	4	5	6	7	8	9	10	11	...
1	1	2	3	5	8	13	21	34	55	89	...

```
public int fib (int n) {  
    if (n == 1 || n == 2)  
        return 1;  
    else  
        return fib (n - 1) + fib (n - 2);  
}
```

```
System.out.println(fib(5));
```

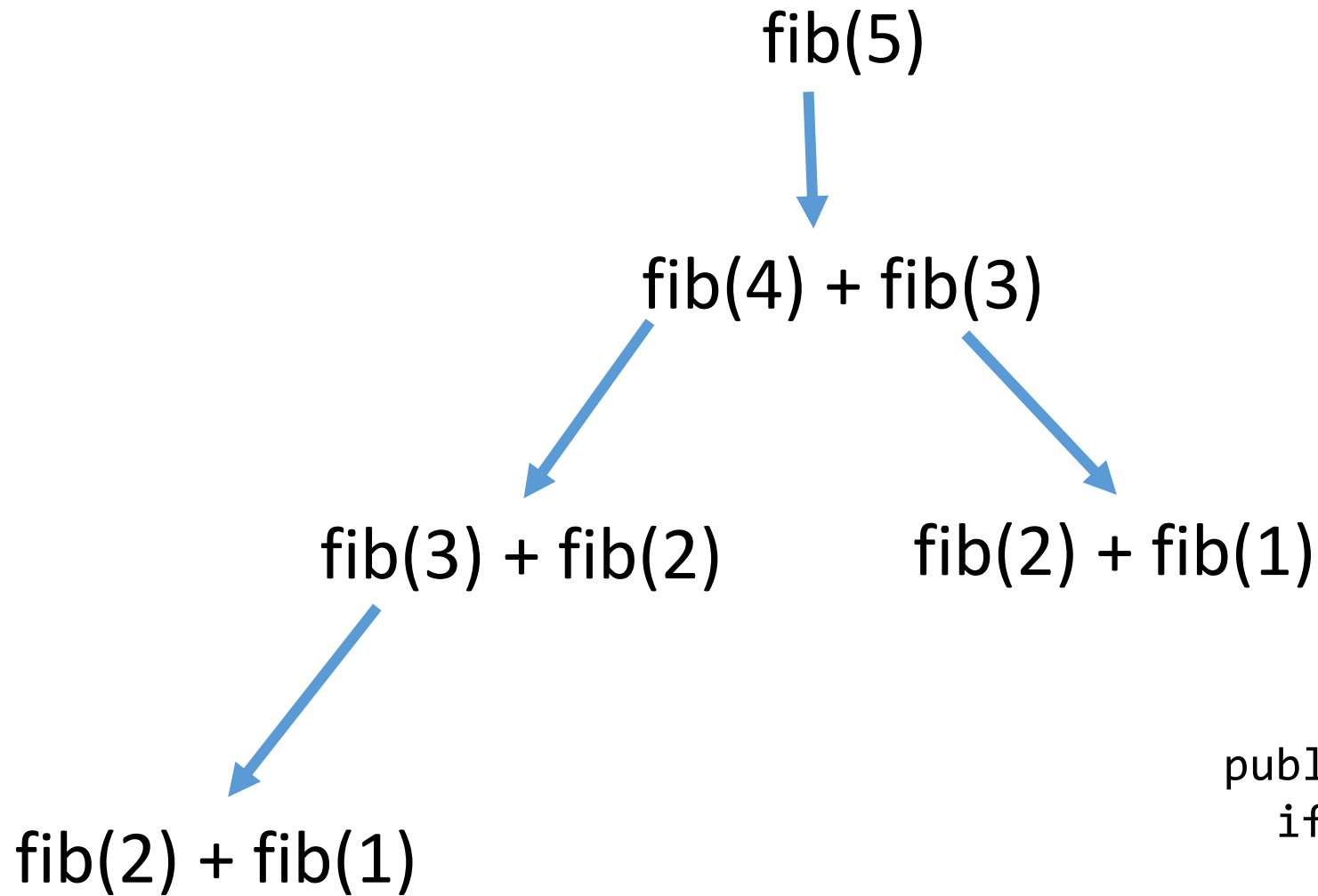
1	2	3	4	5	6	7	8	9	10	11	...
1	1	2	3	5	8	13	21	34	55	89	...



```
public int fib (int n) {  
    if (n == 1 || n == 2)  
        return 1;  
    else  
        return fib (n - 1) + fib (n - 2);  
}
```

```
System.out.println(fib(5));
```

1	2	3	4	5	6	7	8	9	10	11	...
1	1	2	3	5	8	13	21	34	55	89	...

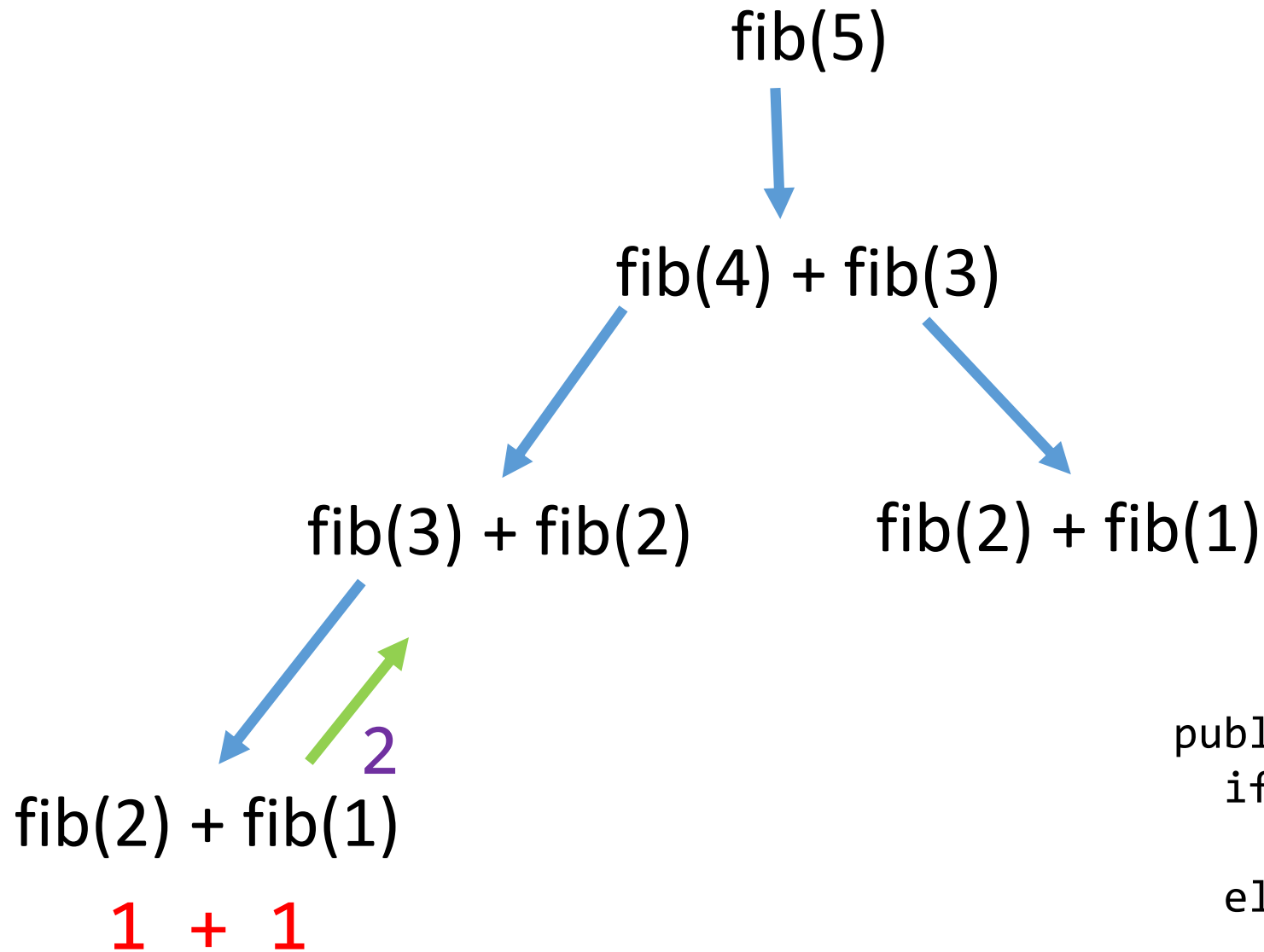


```
public int fib (int n) {  
    if (n == 1 || n == 2)  
        return 1;  
    else  
        return fib (n - 1) + fib (n - 2);  
}
```



```
System.out.println(fib(5));
```

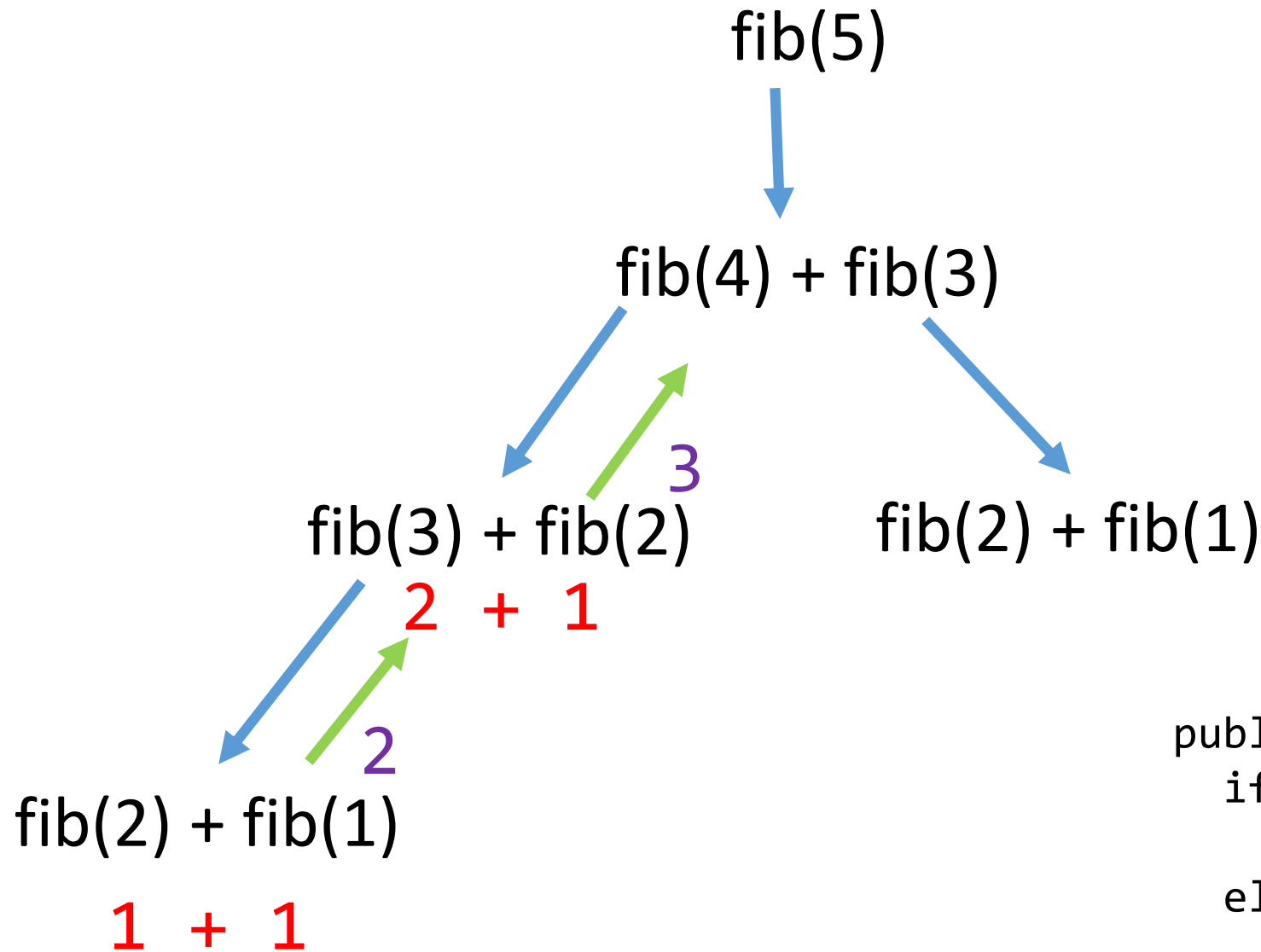
1	2	3	4	5	6	7	8	9	10	11	...
1	1	2	3	5	8	13	21	34	55	89	...



```
public int fib (int n) {  
    if (n == 1 || n == 2)  
        return 1;  
    else  
        return fib (n - 1) + fib (n - 2);  
}
```

```
System.out.println(fib(5));
```

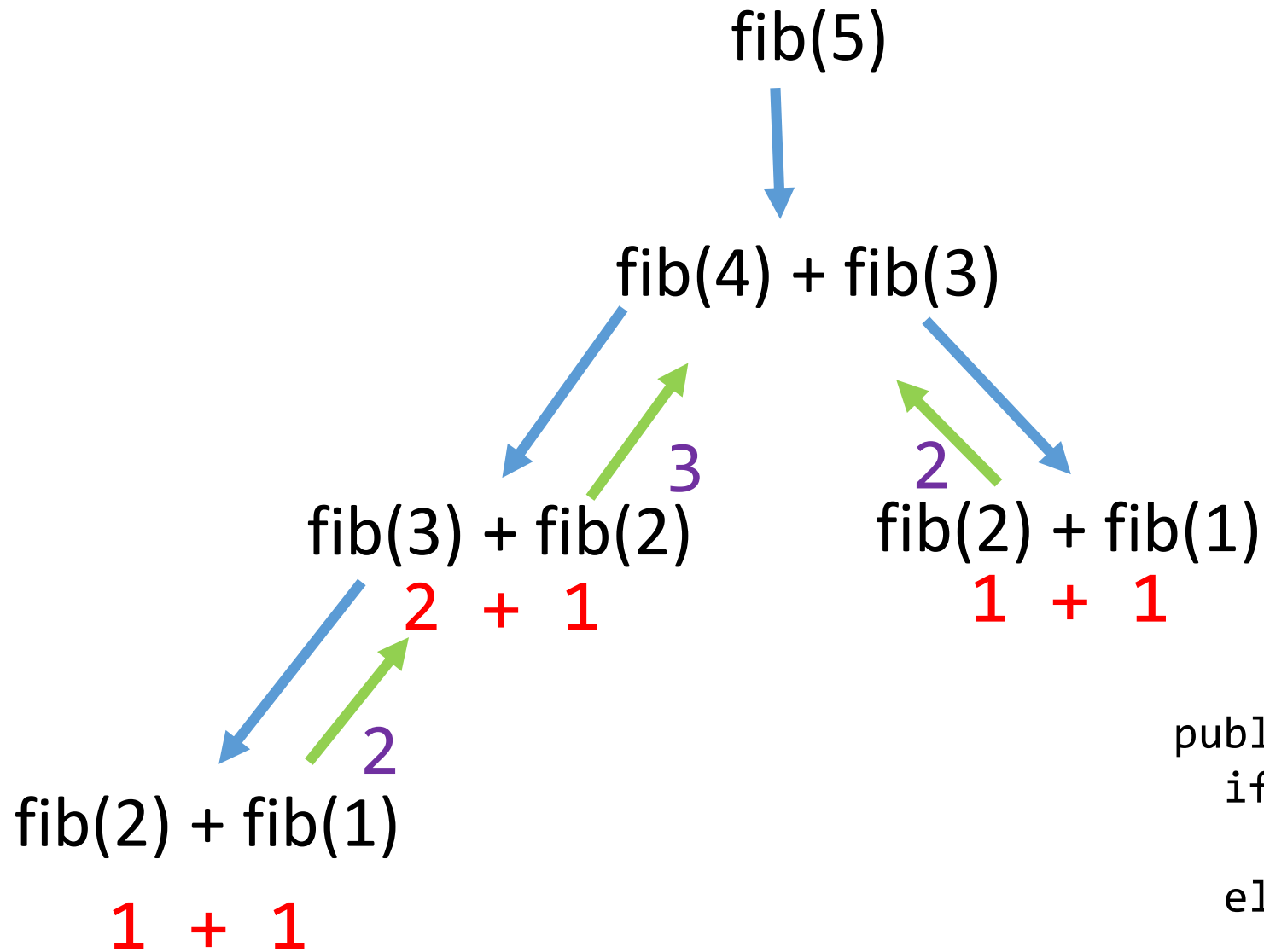
1	2	3	4	5	6	7	8	9	10	11	...
1	1	2	3	5	8	13	21	34	55	89	...



```
public int fib (int n) {  
    if (n == 1 || n == 2)  
        return 1;  
    else  
        return fib (n - 1) + fib (n - 2);  
}
```

```
System.out.println(fib(5));
```

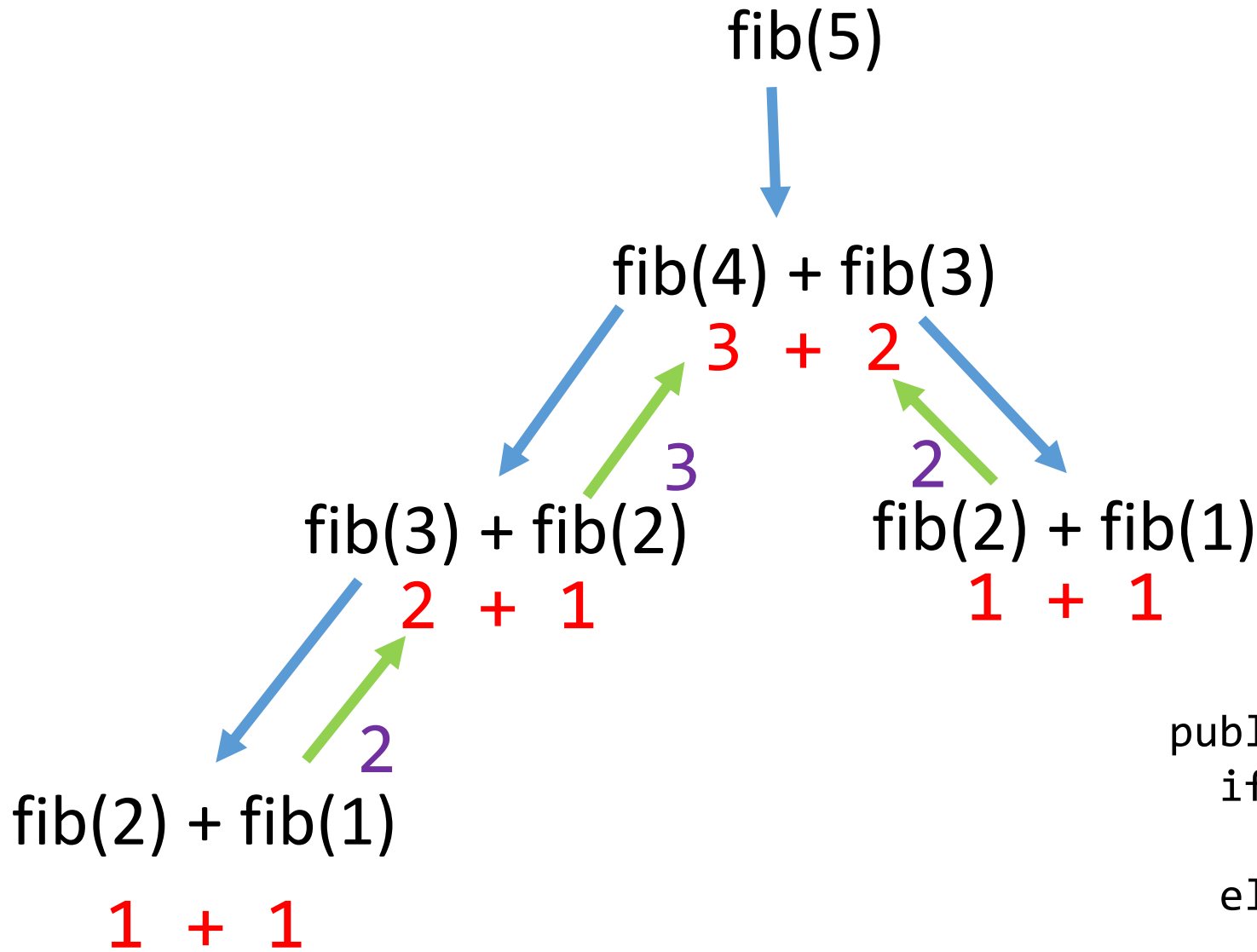
1	2	3	4	5	6	7	8	9	10	11	...
1	1	2	3	5	8	13	21	34	55	89	...



```
public int fib (int n) {  
    if (n == 1 || n == 2)  
        return 1;  
    else  
        return fib (n - 1) + fib (n - 2);  
}
```

```
System.out.println(fib(5));
```

1	2	3	4	5	6	7	8	9	10	11	...
1	1	2	3	5	8	13	21	34	55	89	...

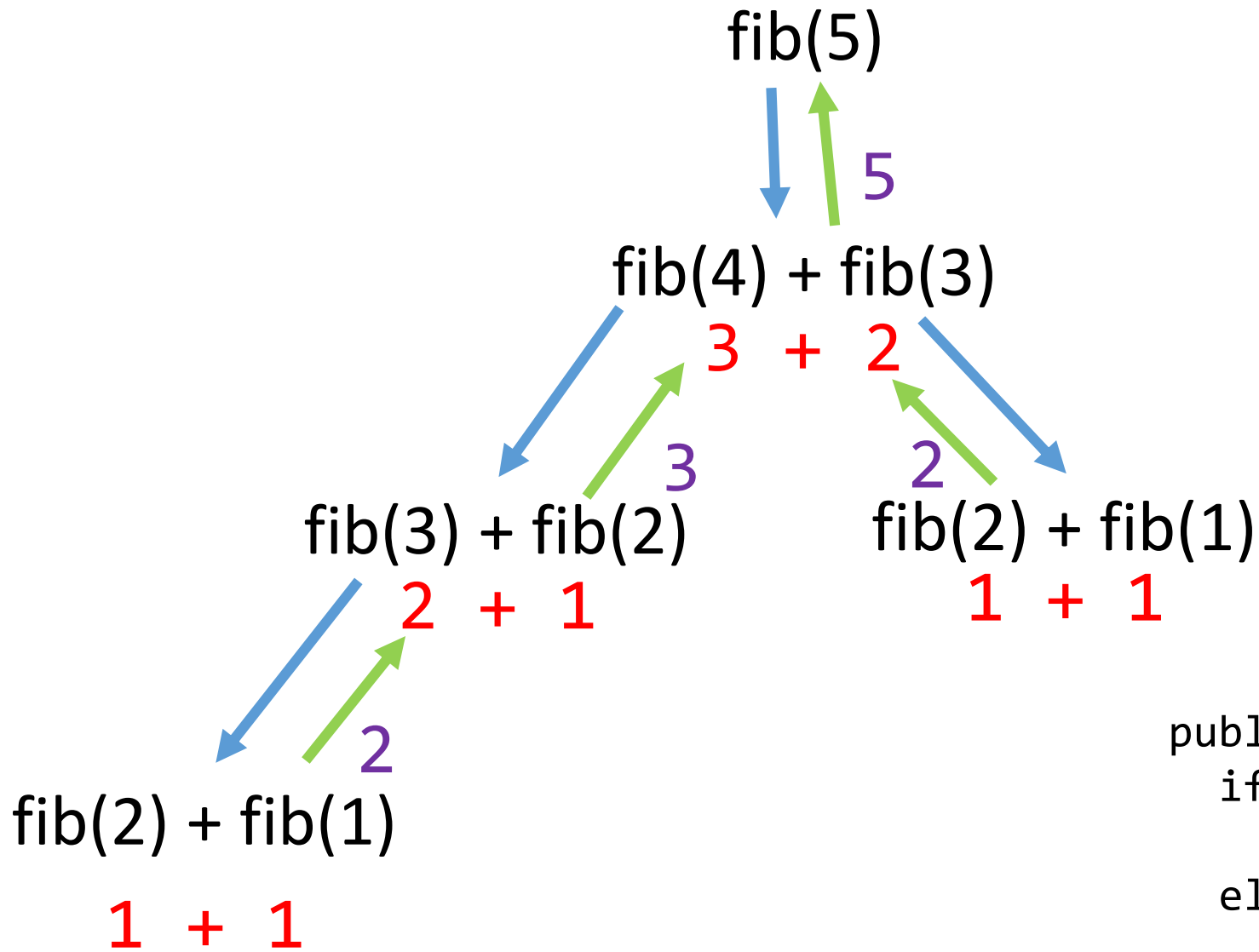


```
public int fib (int n) {  
    if (n == 1 || n == 2)  
        return 1;  
    else  
        return fib (n - 1) + fib (n - 2);  
}
```

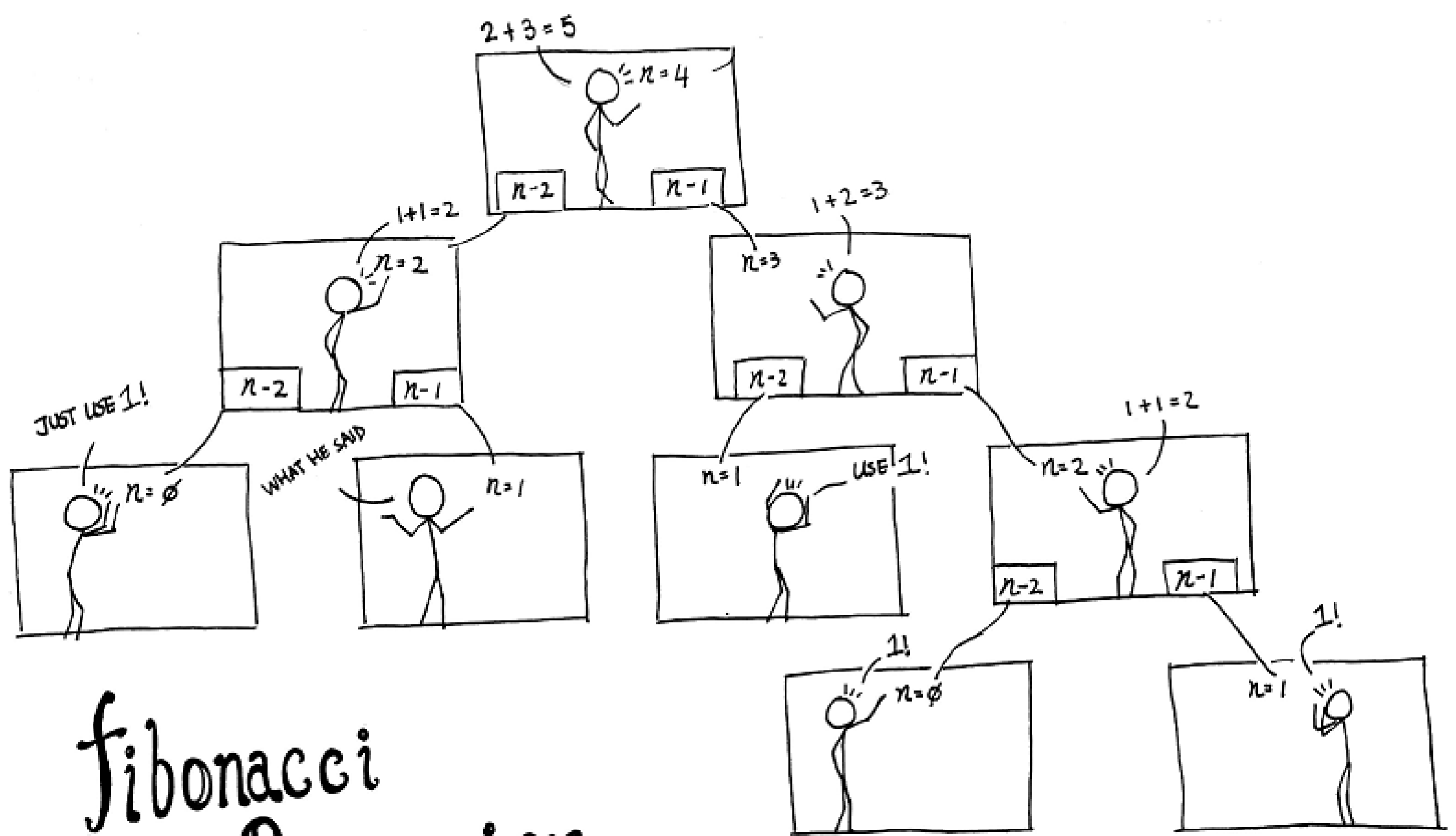


```
System.out.println(fib(5));
```

1	2	3	4	5	6	7	8	9	10	11	...
1	1	2	3	5	8	13	21	34	55	89	...



```
public int fib (int n) {  
    if (n == 1 || n == 2)  
        return 1;  
    else  
        return fib (n - 1) + fib (n - 2);  
}
```



Fibonacci Recursion

DL 2017

TO-DO LIST

1. Make a to-do list