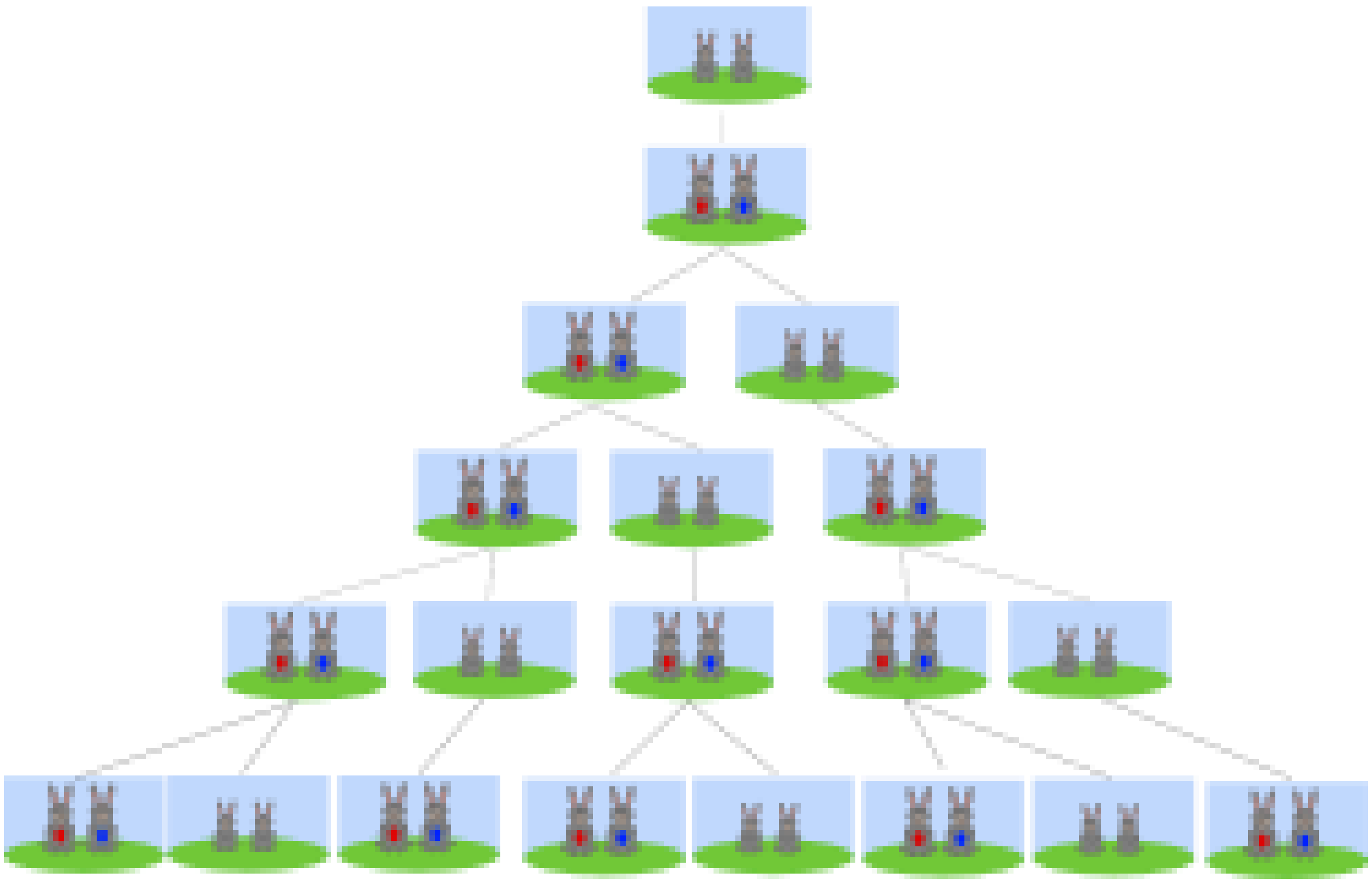
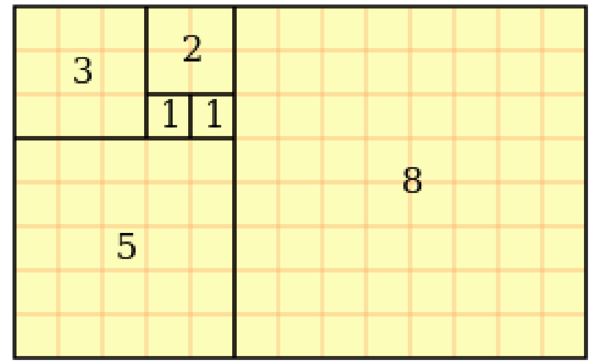
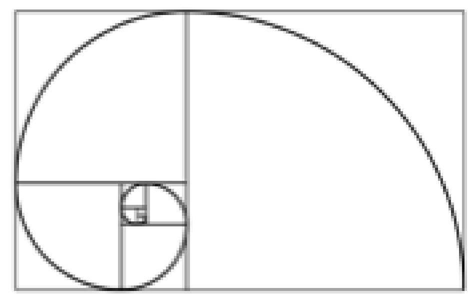
Fibonnacci Number Series

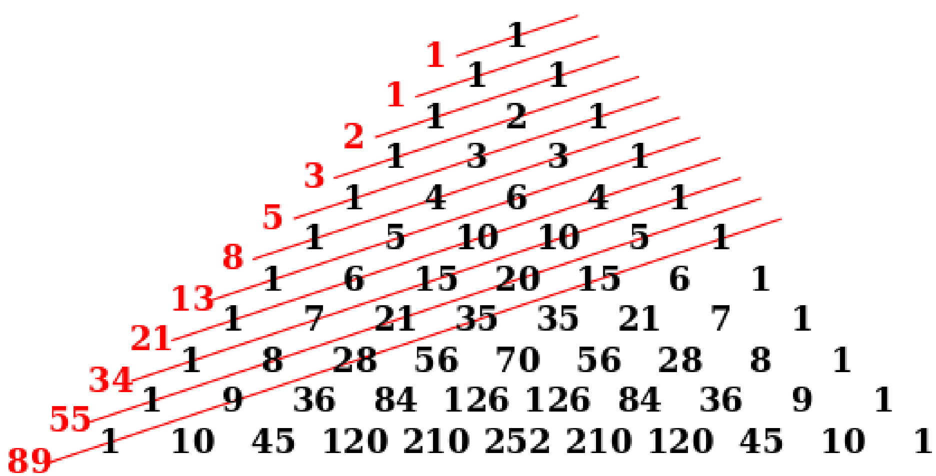


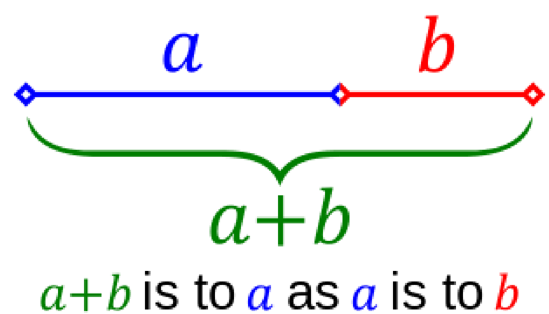
Recursion: Fibonacci

Digression: The Fibonacci Numbers have a long history; the earliest mention is in an analysis of Sanskrit poetry, c. 200 AD, but the name comes from Leonardo of Pisa, better known as Fibonacci, who invented them to explain the geometric growth of a family of rabbits. It has many interesting mathematical properties......







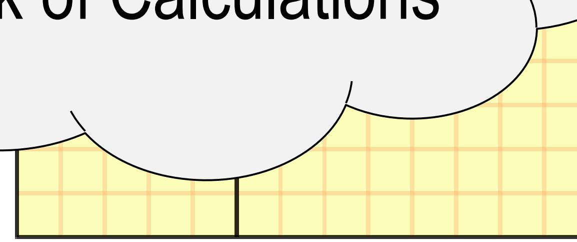


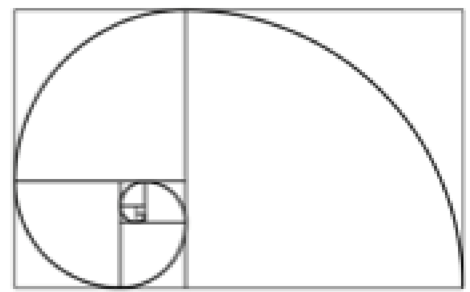
Golden Ratio: 1.6180339....

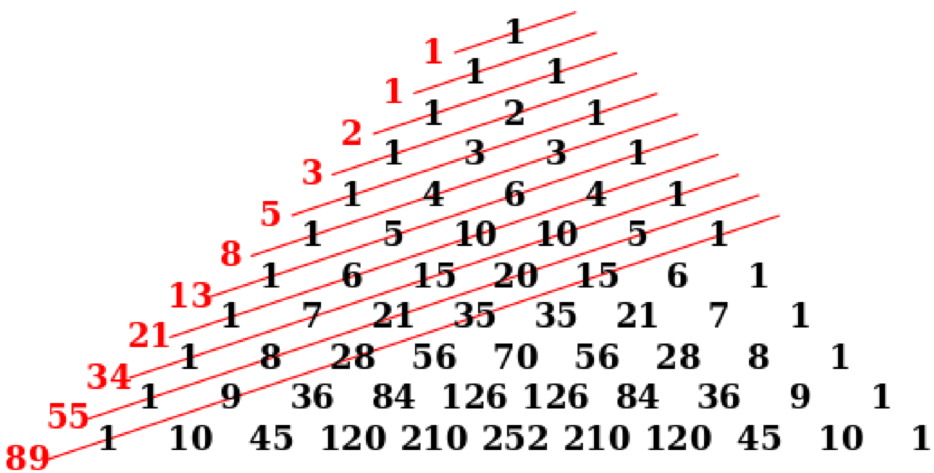
Recursion
F

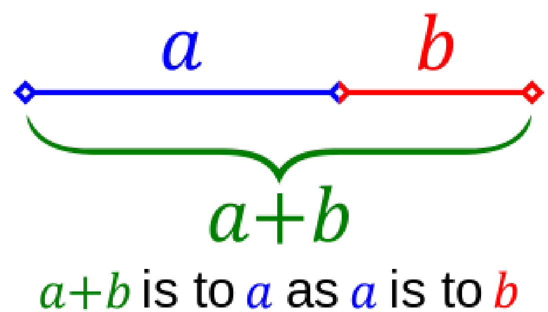
In 1202 wrote
Liber Abaci or
Book of Calculations

Digression: The Fibonacci Numbers ha a long history; the earliest mention is in an analysis of Sanskrit poetry, c. 200 AD, but the name comes from **Leonardo of Pisa**, better known as **Fibonacci**, who invented them to explain the geometric growth of a family of rabbits. It has many interesting mathematical properties......







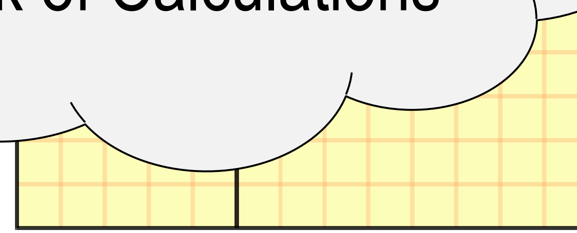


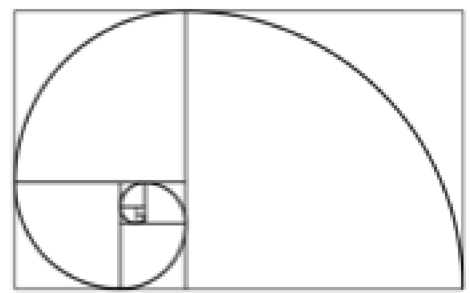
Golden Ratio: 1.6180339....

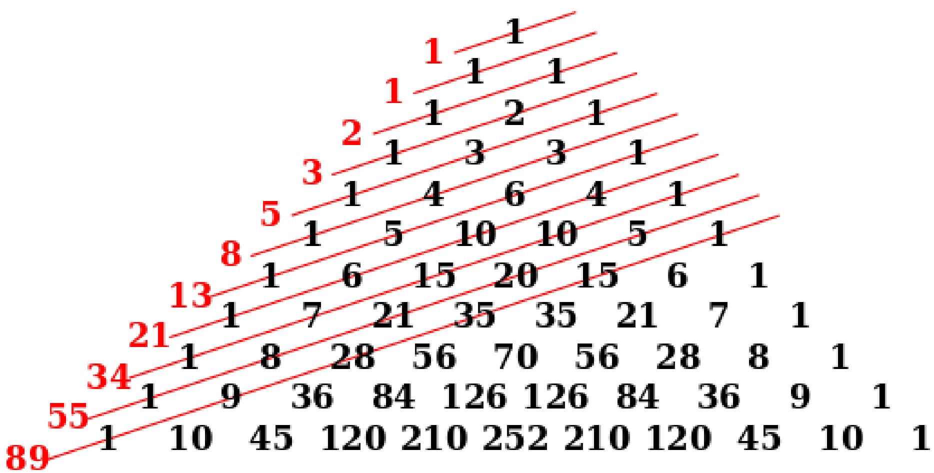
Recursion
F

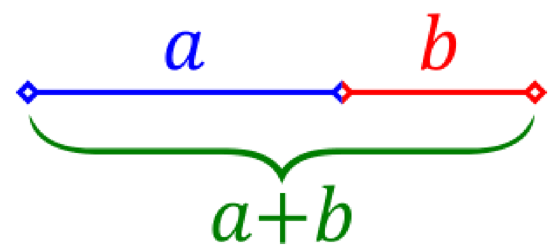
Digression: The Fibonacci Numbers ha
a long history; the earliest mention is in
an analysis of Sanskrit poetry, c. 200 AD,
but the name comes from **Leonardo of Pisa**,
better known as **Fibonacci**, who invented
them to explain the **geometric growth**
of a family of rabbits. It has many interesting
mathematical properties......

In 1202 wrote
Liber Abaci or
Book of Calculations

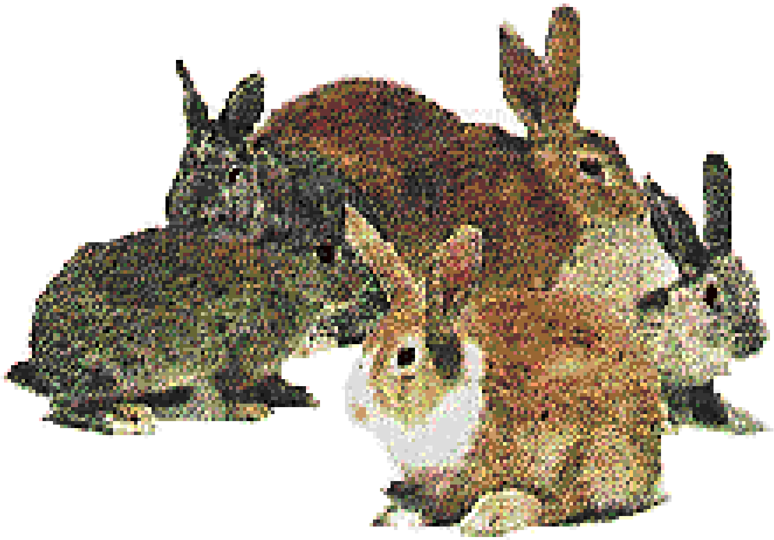








Golden Ratio: 1.6180339....



The original problem that Fibonacci investigated (in the year 1202) was about how fast rabbits could breed in ideal circumstances.

Suppose a newly-born pair of rabbits, one male, one female, are put in a field. Rabbits are able to mate at the age of one month so that at the end of its second month a female can produce another pair of rabbits.

Suppose that our rabbits **never die** and that the female **always** produces one new pair (one male, one female) **every month** from the second month on. The puzzle that Fibonacci posed was...

How many pairs will there be in one year?

Recursion: Fibonacci

Fibonacci Series

Digre

a l

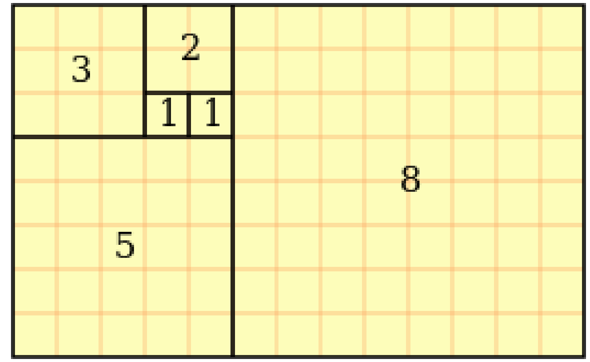
a,

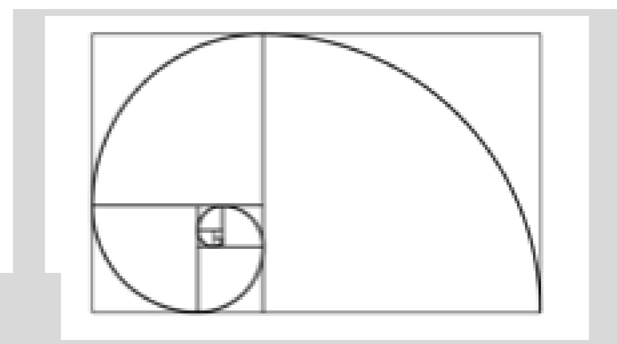
bette ted

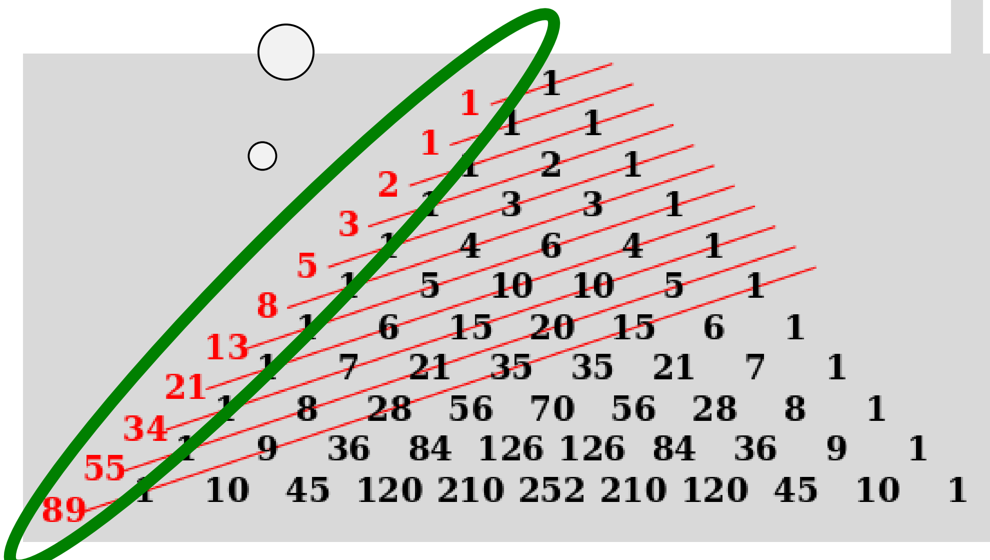
them to explain the **geometric growth**

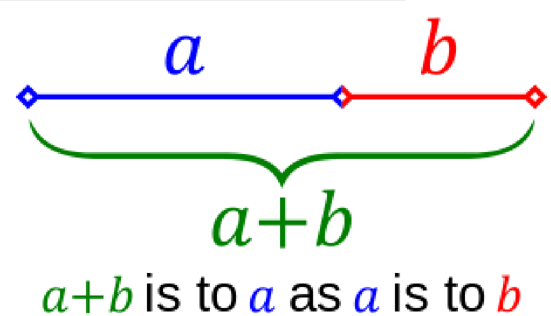
of a family of r bits. It has many interesting

mathematical properties......









Golden Ratio: 1.6180339....

Recursion: Fibonacci

Digre…

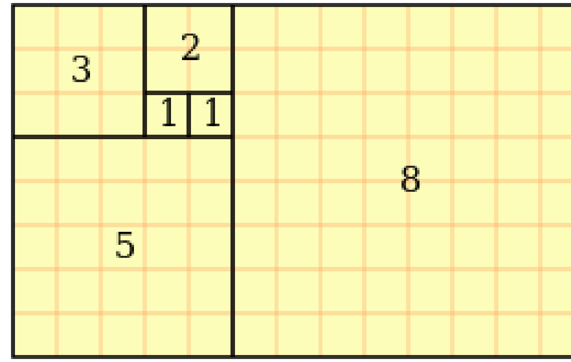
a l…

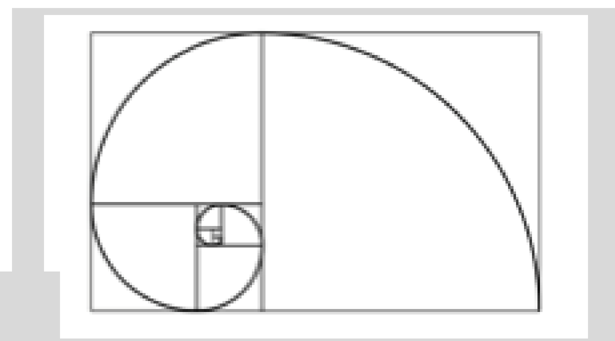
Perfect Spiral

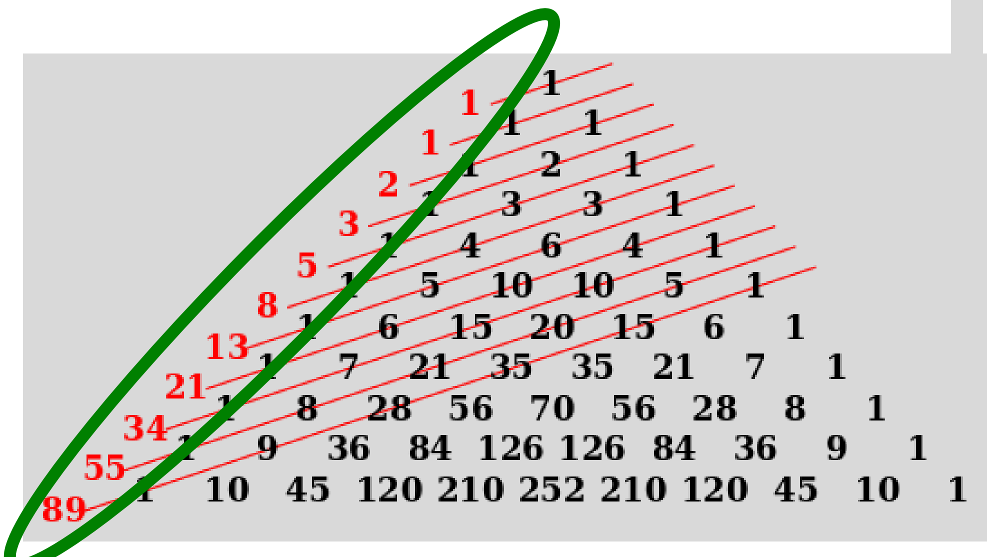
bette… …ed

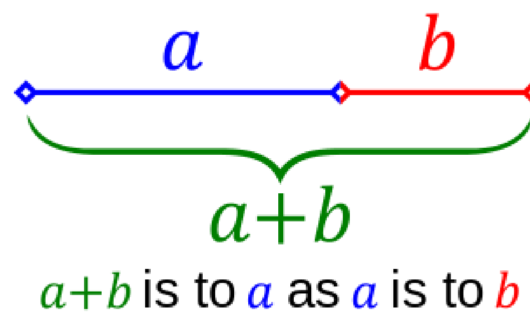
them to explain the g…metric growth

of a family of rabbits. **It has many interesting mathematical properties……**









Golden Ratio: 1.6180339….

Recursion: Fibonacci

Golden Ratio

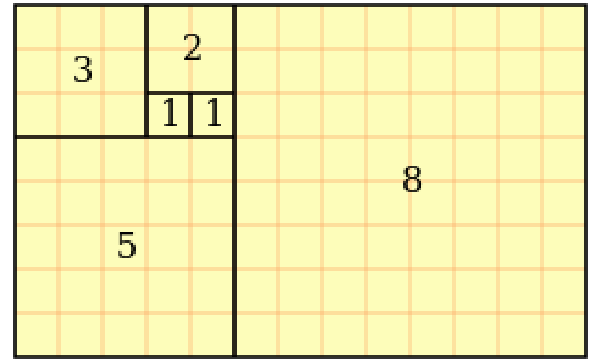
Digre

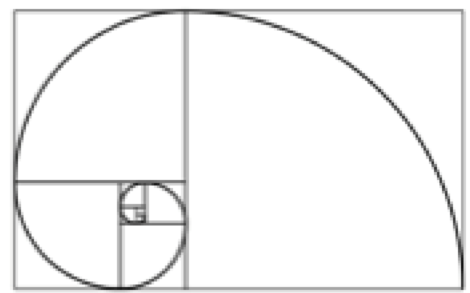
a l

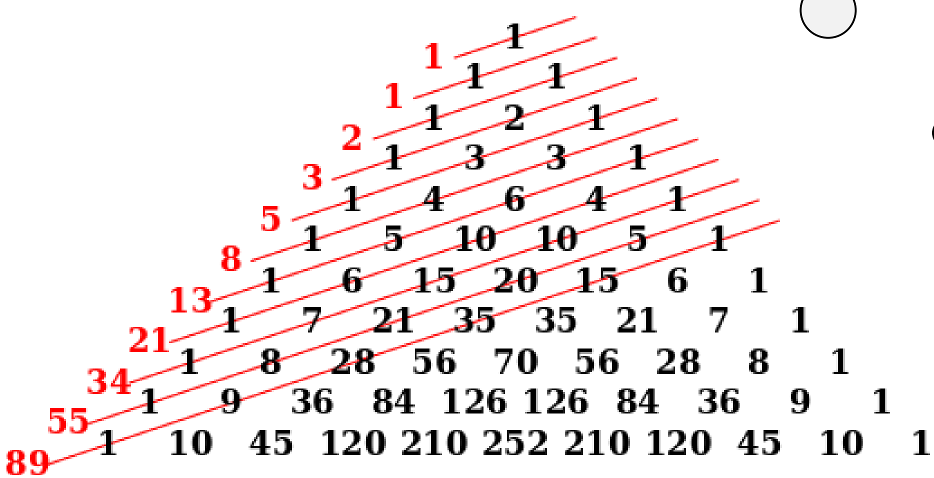
bette ... ed

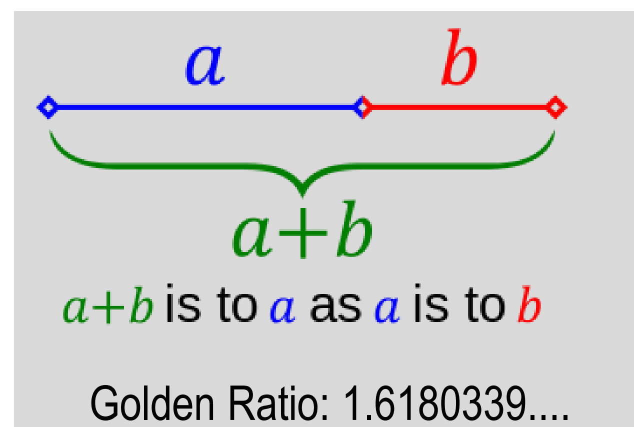
them to explain the ge...tric growth

of a family of rabbits. **It has many in...esting mathematical properties......**









Recursion:
Fibonacci

Let's look at another example, the Fibonacci numbers, defined (two ways) as follows:

F = { 1, 1, 2, 3, 5, **8**, 13, 21, ..., **(sum of previous 2 terms)**, ... }

0 1 2 3 4 **5** 6 7

```
// returns the ith Fibonacci number

int fib(int i) {
   int lim = 2;
   if( i < lim )
      return 1;
   else
      return fib(i-1) + fib(i-2);
}
```

Recursion:
Fibonacci

Let's look at another example, the Fibonacci numbers, defined (two ways) as follows:

F = { 1, 1, 2, 3, **5**, 8, 13, 21, …., **(sum of previous 2 terms)**, … }

0 1 2 3 **4** **5** 6 7

```
// returns the ith Fibonacci number

int fib(int i) {
   int lim = 2;
   if( i < lim )
      return 1;
   else
      return fib(i-1) + fib(i-2);
}
```

Recursion: Fibonacci

Let's look at another example, the Fibonacci numbers, defined (two ways) as follows:

F = { 1, 1, 2, **3**, 5, 8, 13, 21, ..., **(sum of previous 2 terms)**, ... }

0 1 2 **3** 4 5 6 7

```
// returns the ith Fibonacci number

int fib(int i) {
   int lim = 2;
   if( i < lim )
      return 1;
   else
      return fib(i-1) + fib(i-2);
}
```

Recursion: Fibonacci

Let's look at another example, the Fibonacci numbers, defined (two ways) as follows:

F = { 1, 1, **2**, 3, 5, 8, 13, 21, ..., (sum of previous 2 terms), ... }

0 1 **2** 3 4 5 6 7

```
// returns the ith Fibonacci number

int fib(int i) {
   int lim = 2;
   if( i < lim )
      return 1;
   else
      return fib(i-1) + fib(i-2);
}
```

Recursion: Fibonacci

Let’s look at another example, the Fibonacci numbers, defined (two ways) as follows:

F = { **1**, **1**, 2, 3, 5, 8, 13, 21, ..., (sum of previous 2 terms), ... }

0 **1** 2 3 4 5 6 7

```
// returns the ith Fibonacci number

int fib(int i) {
   int lim = 2;
   if( i < lim )
      return 1;
   else
      return fib(i-1) + fib(i-2);
}
```

Recursion:
Fibonacci

Let’s look at another example, the Fibonacci numbers, defined (two ways) as follows:

F = { 1, 1, 2, 3, 5, 8, 13, 21, …., **(sum of previous 2 terms)**, … }

0 1 2 3 4 5 6 7

```
// returns the ith Fibonacci number

int fib(int i) {
   int lim = 2;
   if( i < lim )
      return 1;
   else
      return fib(i-1) + fib(i-2);
}
```

Recursion: Fibonacci

Let's look at another example, the Fibonacci numbers, defined (two ways) as follows:

F = { 1, 1, 2, 3, 5, 8, 13, 21, ..., **(sum of previous 2 terms)**, ... }

0 1 2 3 4 5 6 7

```
// returns the ith Fibonacci number

int fib(int i) {
   int lim = 2;
   if( i < lim )
      return 1;
   else
      return fib(i-1) + fib(i-2);
}
```

Recursion: Fibonacci

Let's look at another example, the Fibonacci numbers, defined (two ways) as follows:

F = { 1, 1, 2, 3, 5, 8, 13, 21, ..., **(sum of previous 2 terms)**, ... }

0 1 2 3 4 5 6 7

```
// returns the ith Fibonacci number

int fib(int i) {
    int lim = 2;
    if( i < lim )
        return 1;
    else
        return fib(i-1) + fib(i-2);
}
```

Recursion:
Fibonacci

Let's look at another example, the Fibonacci numbers, defined (two ways) as follows:

F = { 1, 1, 2, 3, 5, 8, 13, 21, …., **(sum of previous 2 terms)**, … }

0 1 2 3 4 5 6 7

```
// returns the ith Fibonacci number

int fib(int i) {
   int lim = 2;
   if( i < lim )
      return 1;
   else
      return fib(i-1) + fib(i-2);
}
```

Recursion: Fibonacci

Let's look at another example, the Fibonacci numbers, defined (two ways) as follows:

F = { 1, 1, 2, 3, 5, 8, 13, 21, …., **(sum of previous 2 terms)**, … }

0 1 2 3 4 5 6 7

```
// returns the ith Fibonacci number

int fib(int i) {
    int lim = 2;
    if( i < lim )
        return 1;
    else
        return fib(i-1) + fib(i-2);
}
```

Recursion:
Fibonacci

Let's look at another example, the Fibonacci numbers, defined (two ways) as follows:

F = { 1, 1, 2, 3, 5, 8, 13, 21, ..., **(sum of previous 2 terms)**, ... }

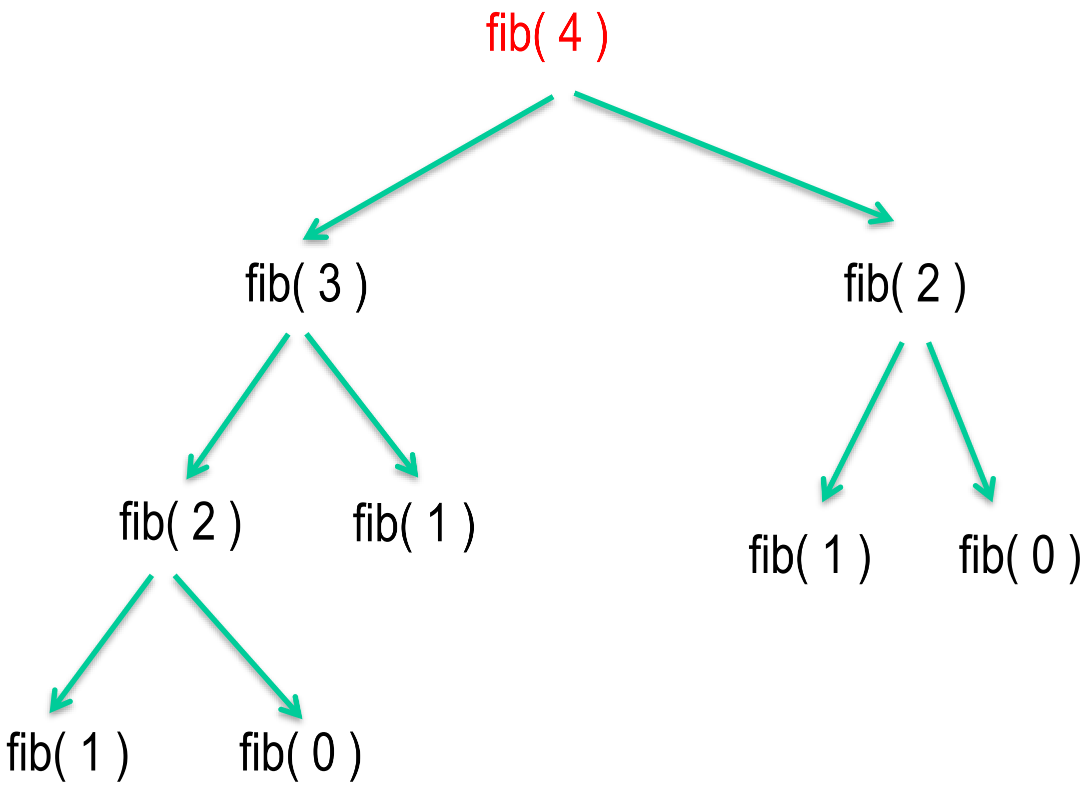
0 1 2 3 4 5 6 7

```
// returns the ith Fibonacci number

int fib(int i) {
    int lim = 2;
    if( i < lim )
        return 1;
    else
        return fib(i-1) + fib(i-2);
}
```

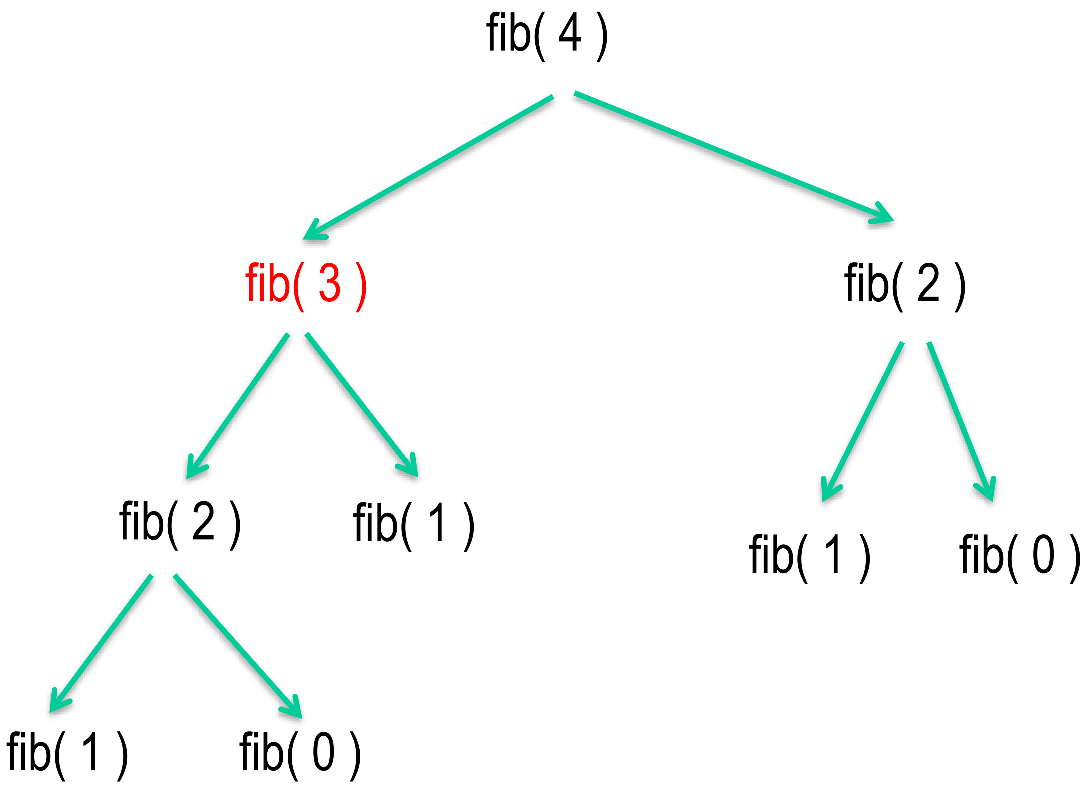
Recursion: Fibonacci

Tree of Calls to fib(4)



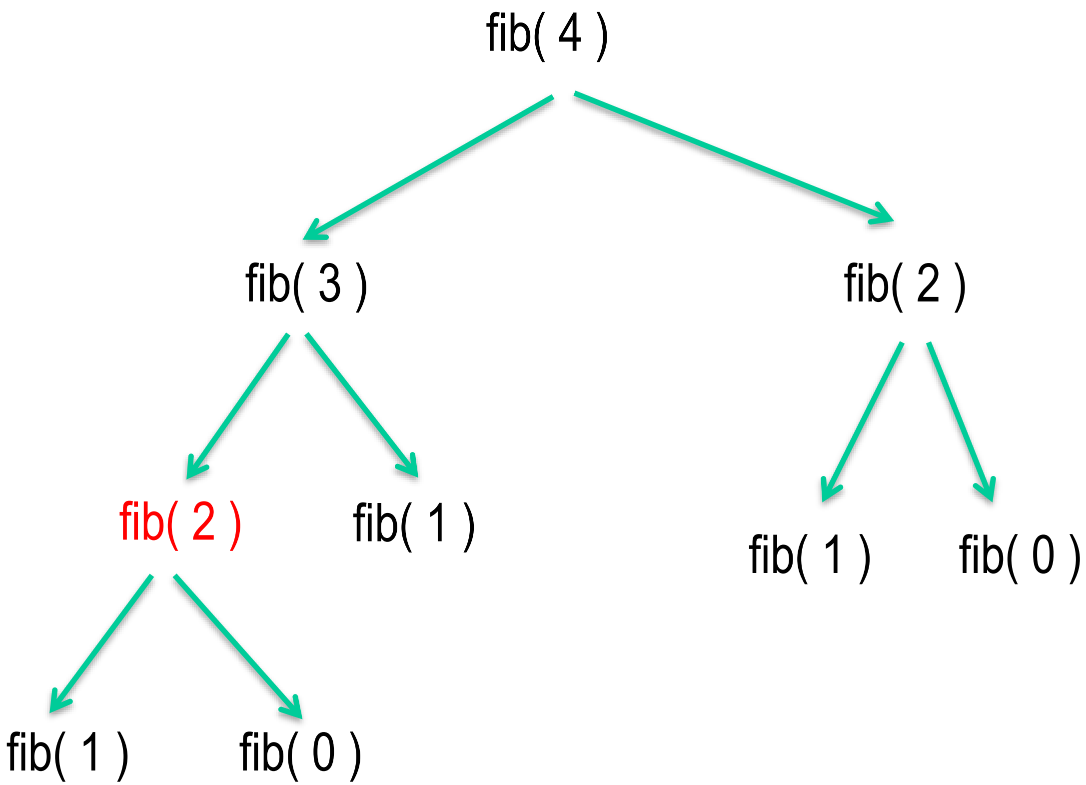
Recursion: Fibonacci

Tree of Calls to fib(4)



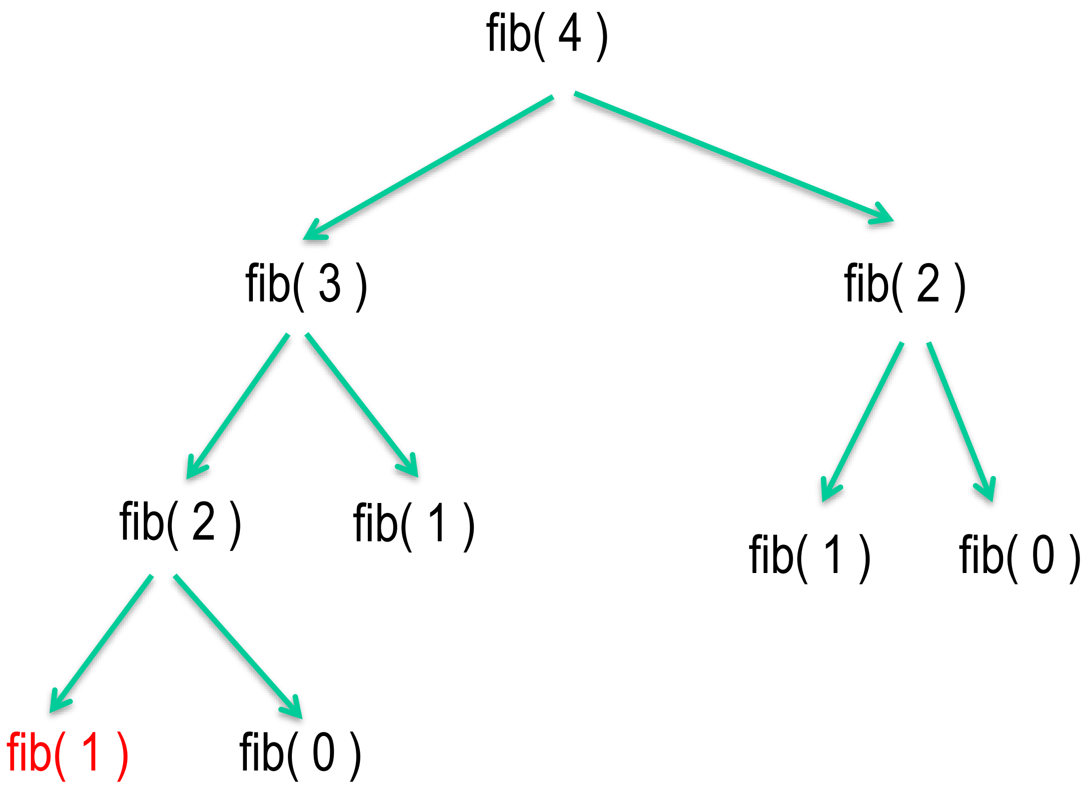
Recursion: Fibonacci

Tree of Calls to fib(4)



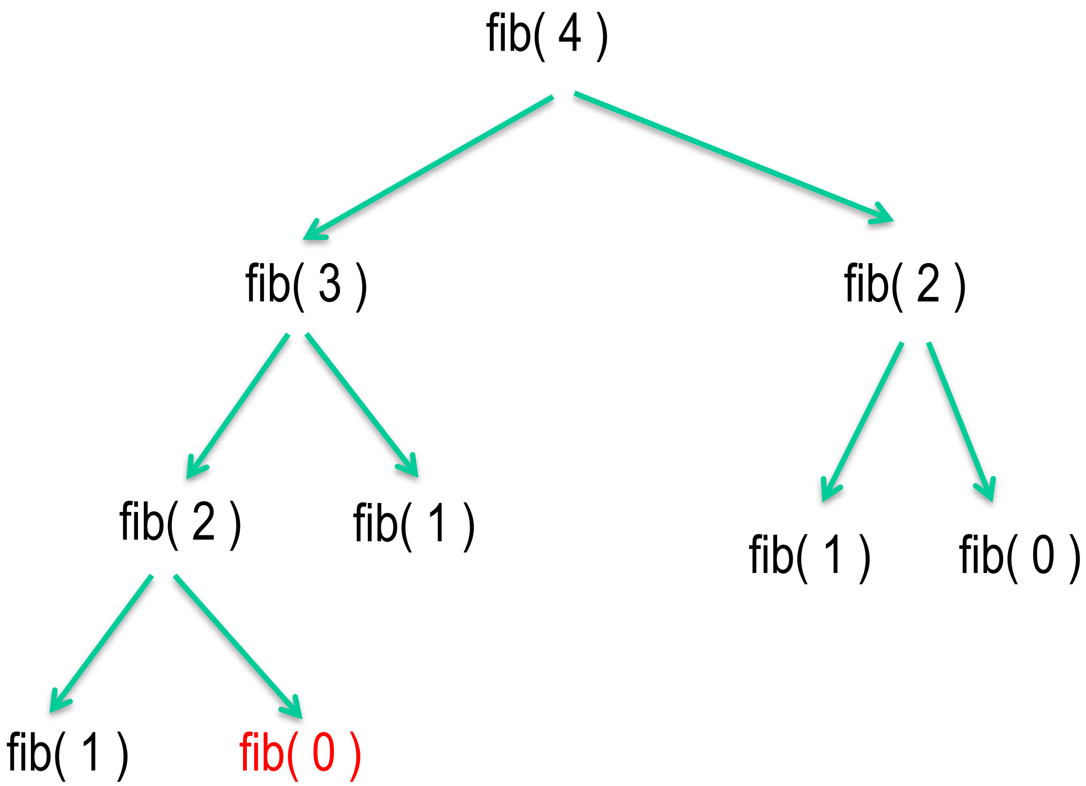
Recursion: Fibonacci

Tree of Calls to fib(4)



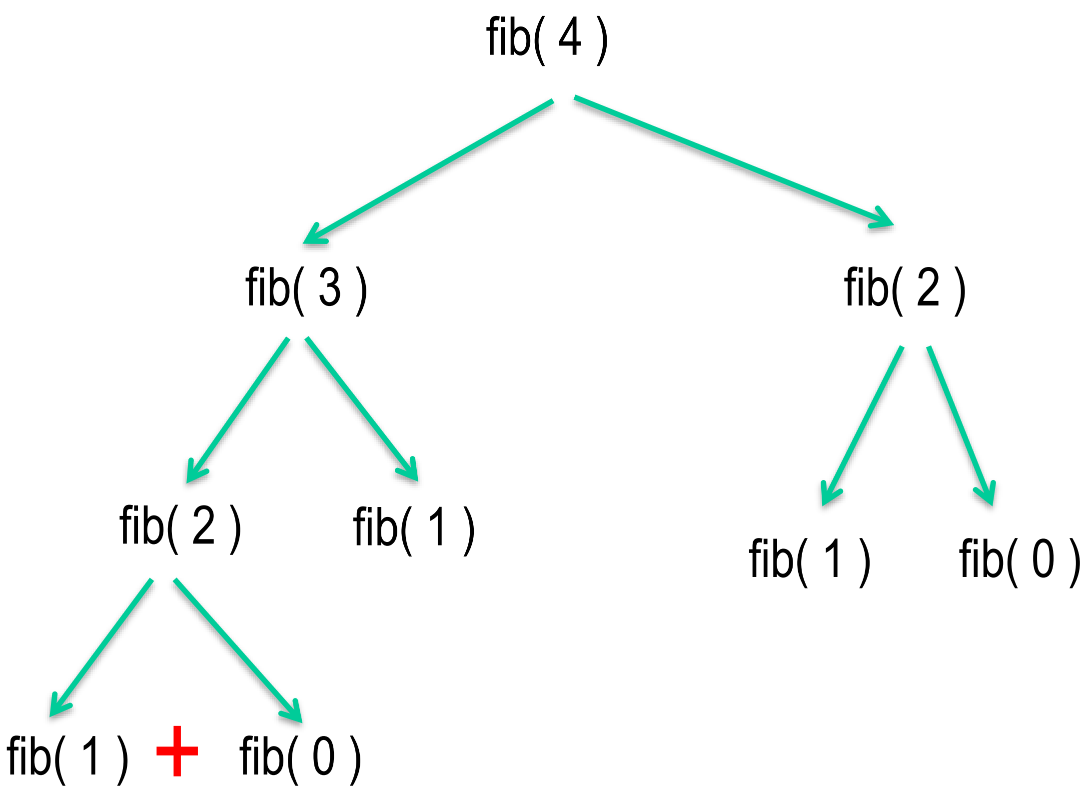
Recursion: Fibonacci

Tree of Calls to fib(4)



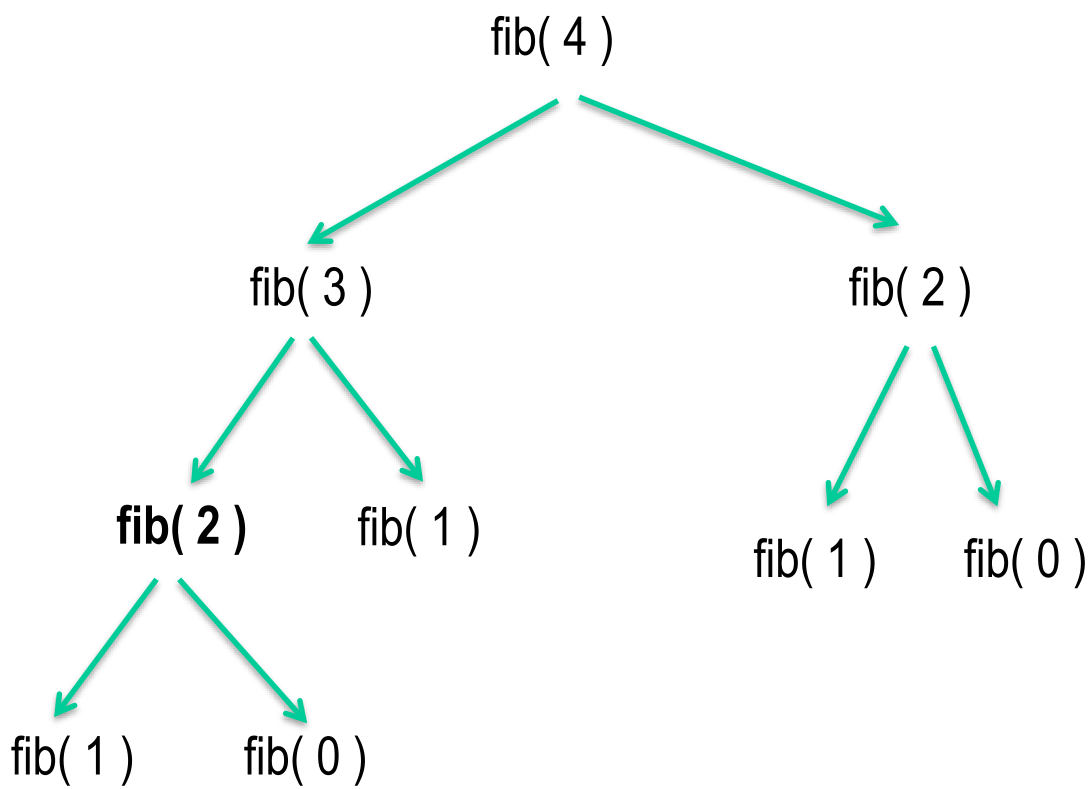
Recursion: Fibonacci

Tree of Calls to fib(4)



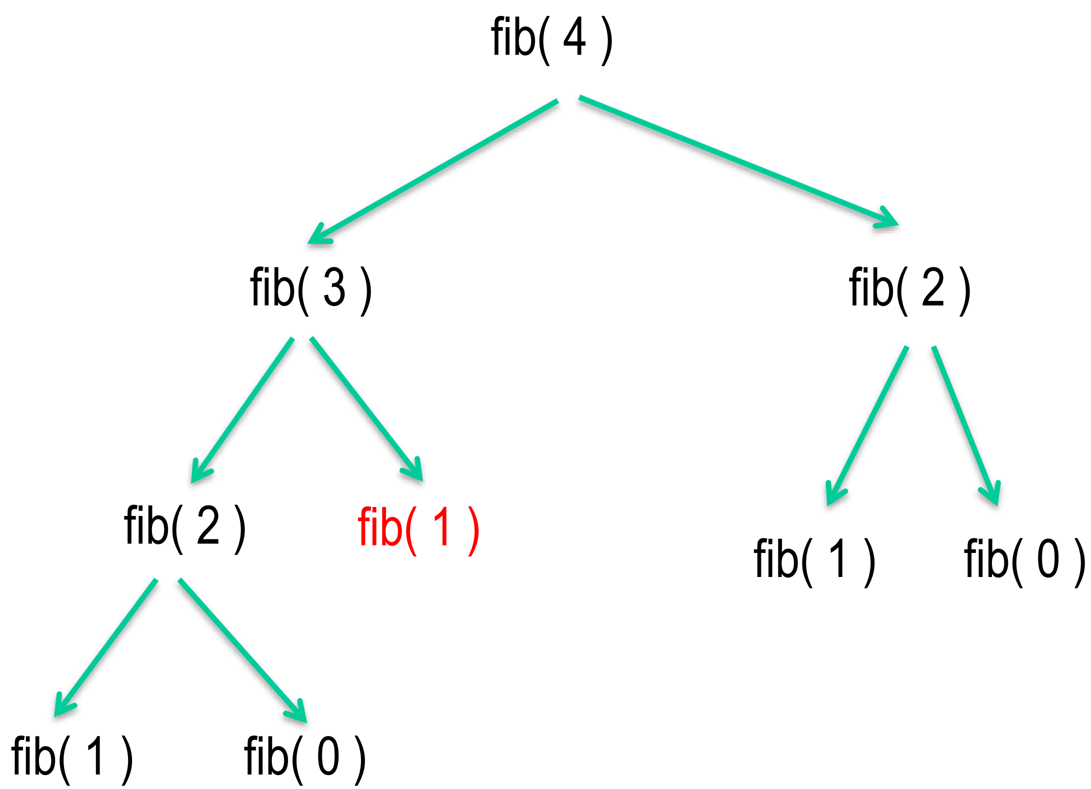
Recursion: Fibonacci

Tree of Calls to fib(4)



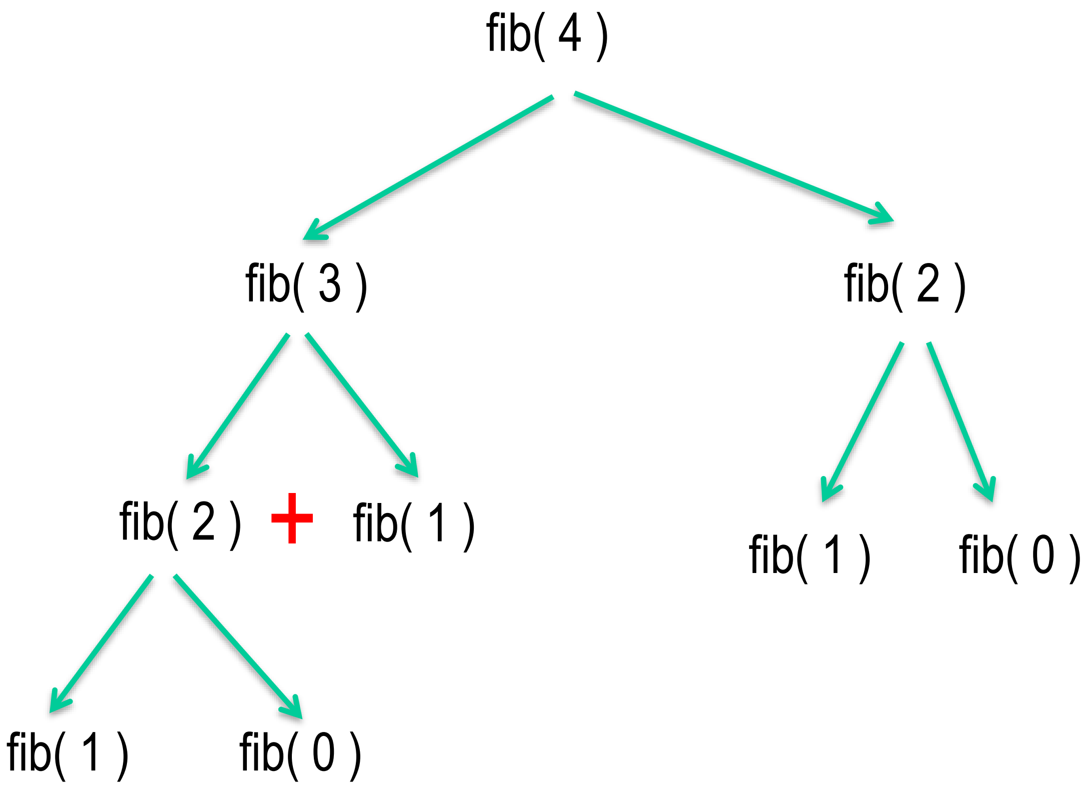
Recursion: Fibonacci

Tree of Calls to fib(4)



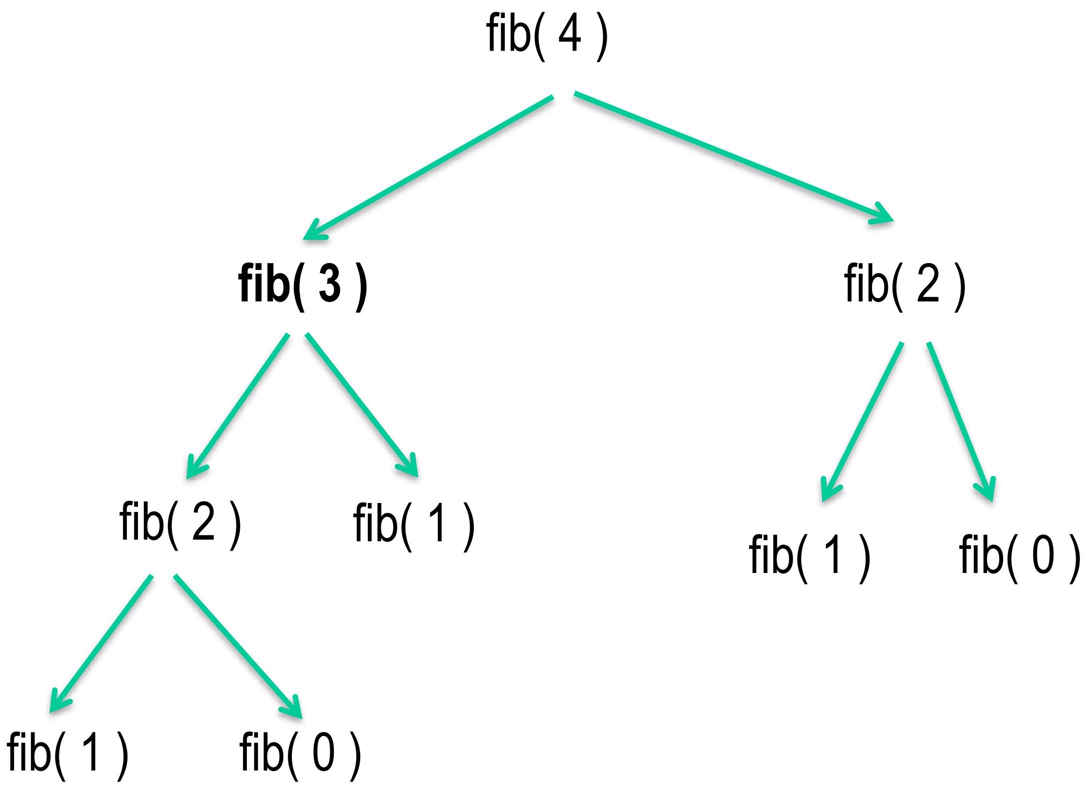
Recursion: Fibonacci

Tree of Calls to fib(4)



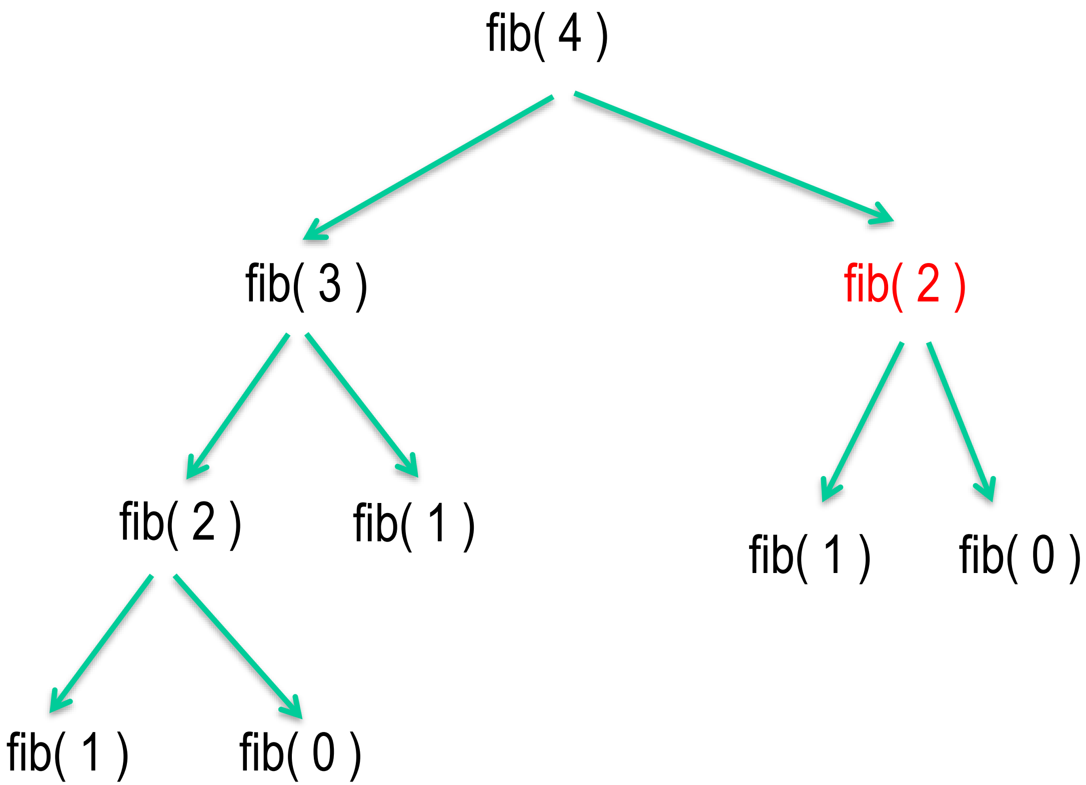
Recursion: Fibonacci

Tree of Calls to fib(4)



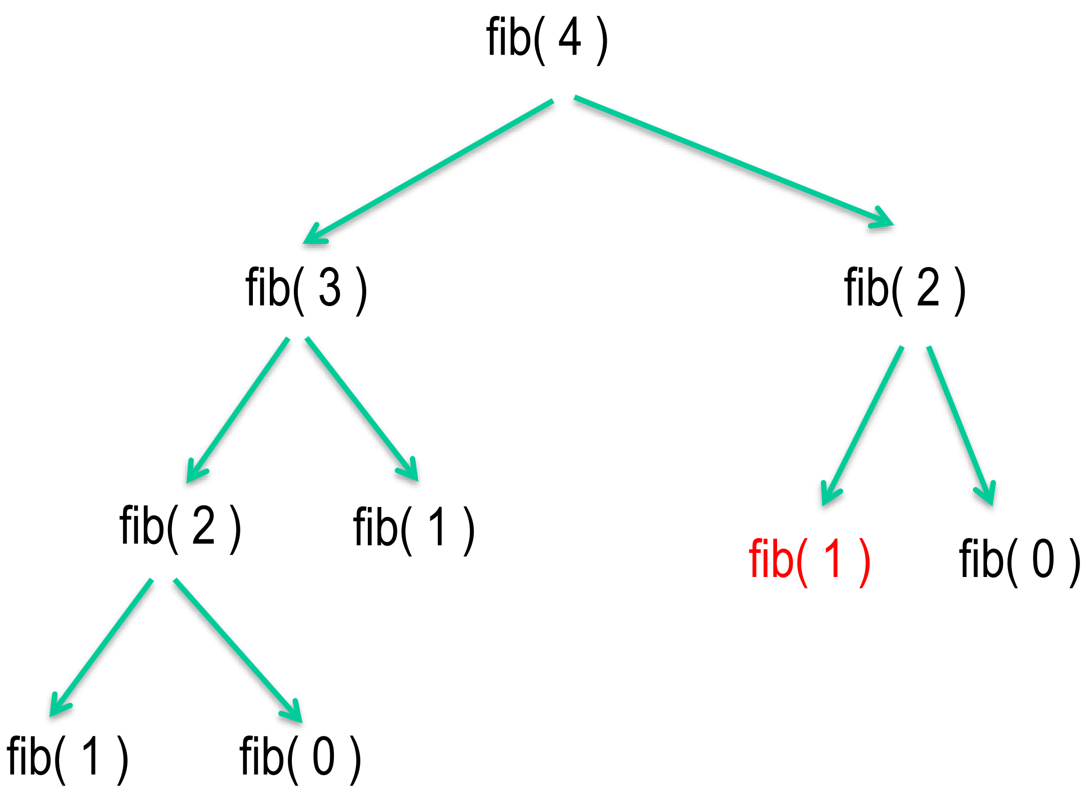
Recursion: Fibonacci

Tree of Calls to fib(4)



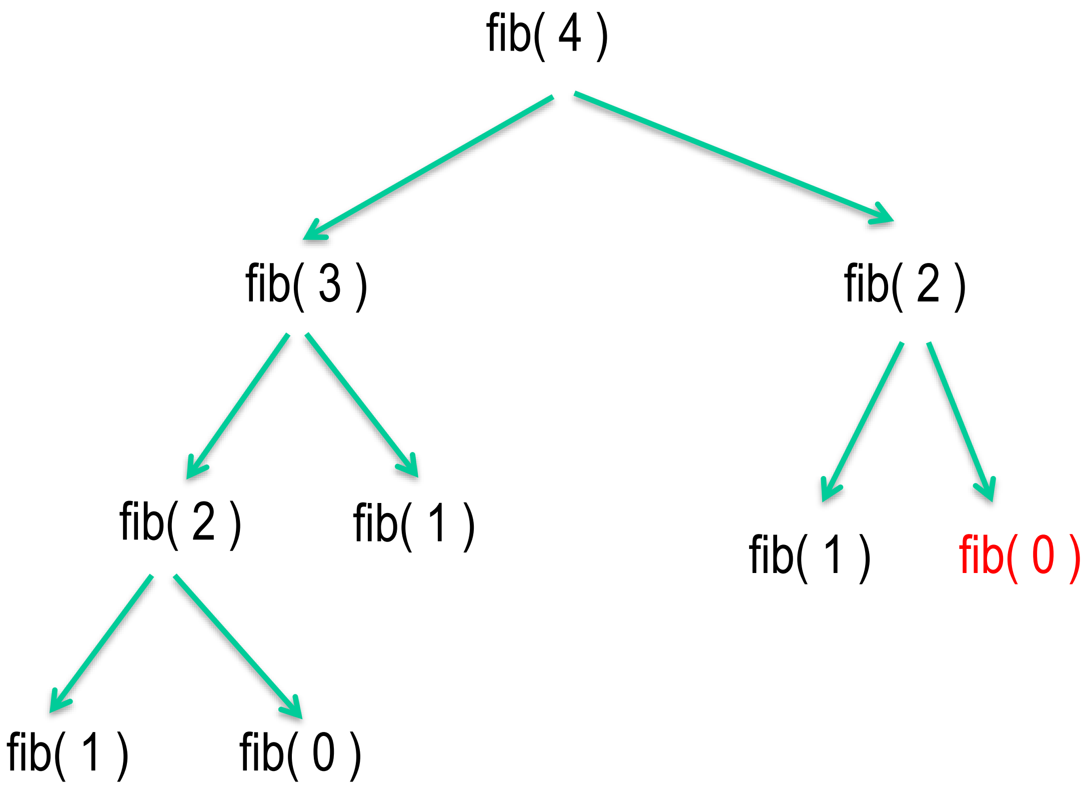
Recursion: Fibonacci

Tree of Calls to fib(4)



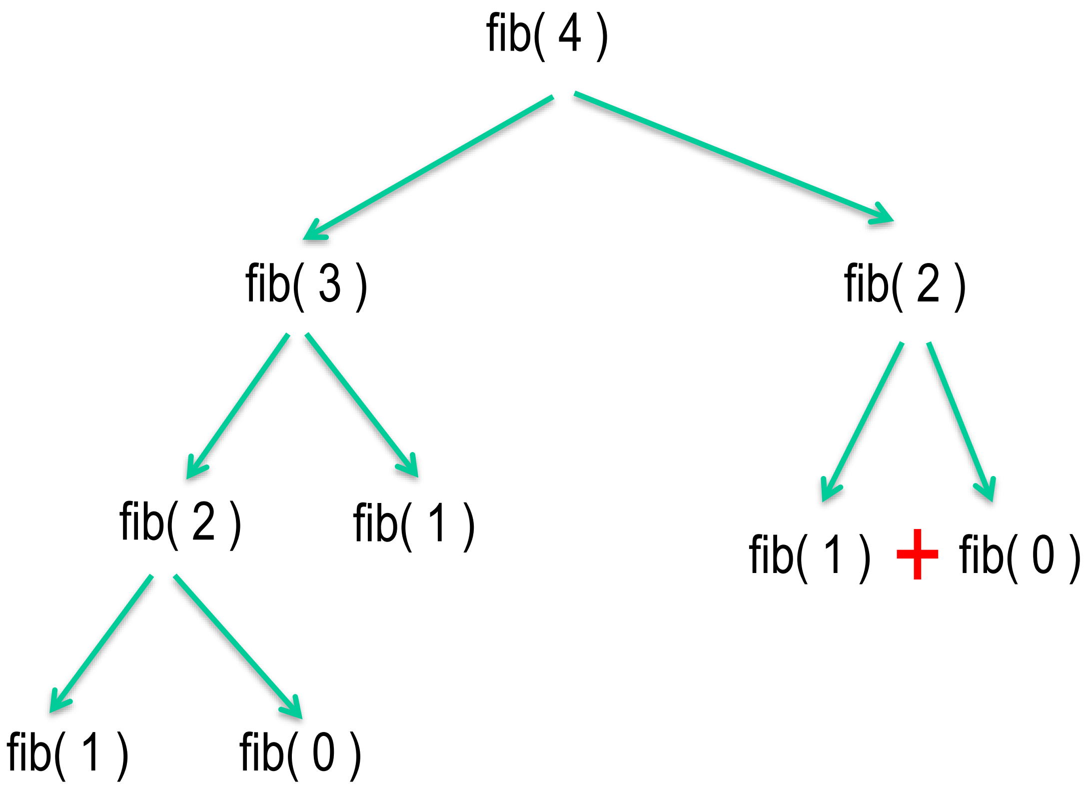
Recursion: Fibonacci

Tree of Calls to fib(4)



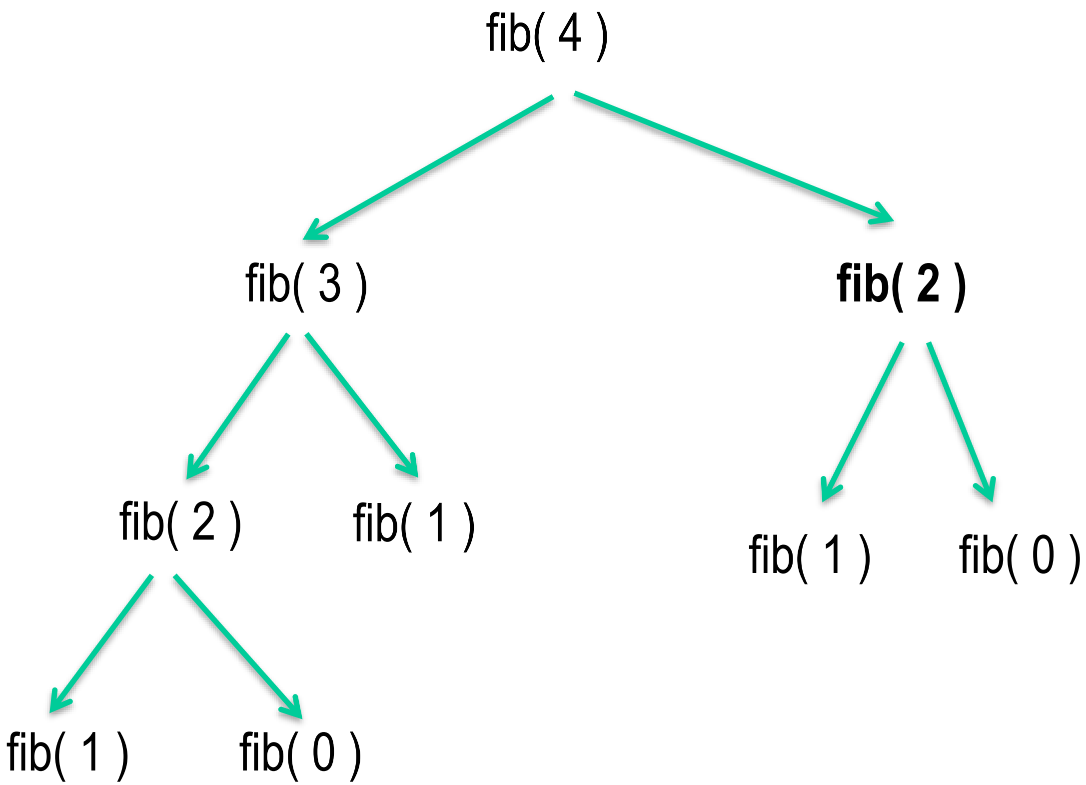
Recursion: Fibonacci

Tree of Calls to fib(4)



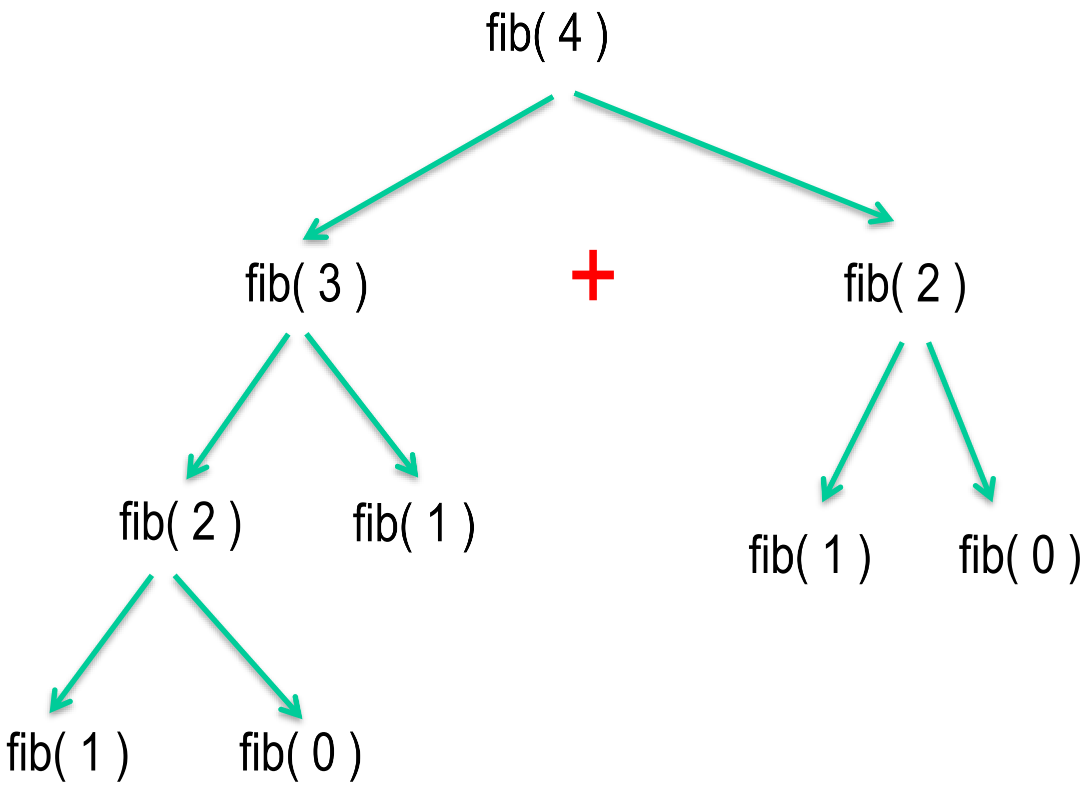
Recursion: Fibonacci

Tree of Calls to fib(4)



Recursion: Fibonacci

Tree of Calls to fib(4)



Recursion: Fibonacci

Tree of Calls to fib(4)