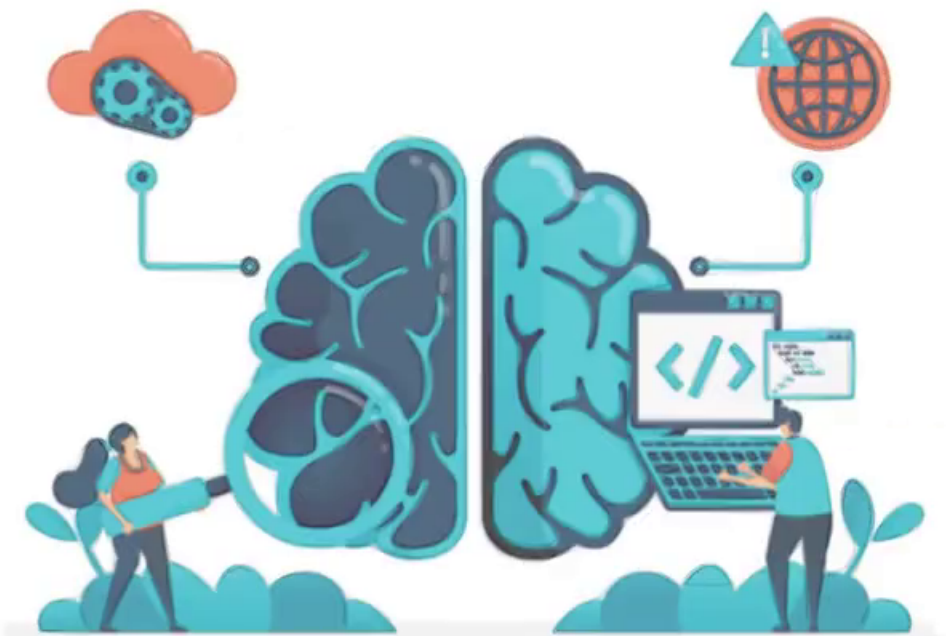
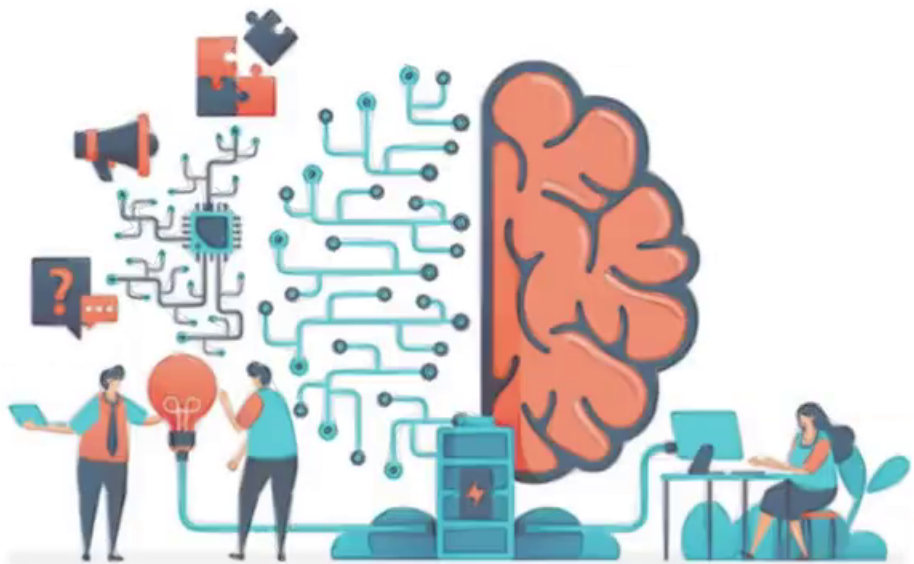
HOW DOES RECURSION REMEMBER STUFF ? (ARABIC INTUITION)



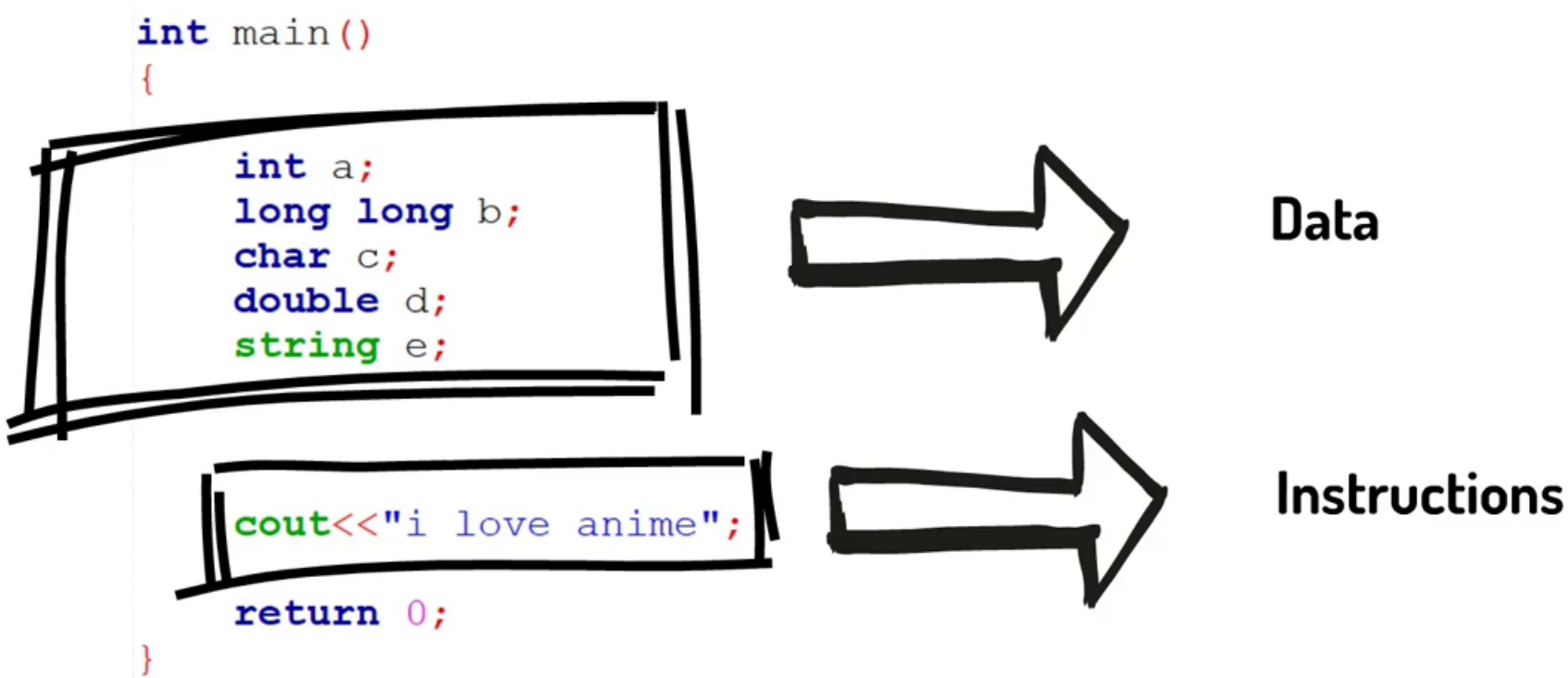


prior knowledge

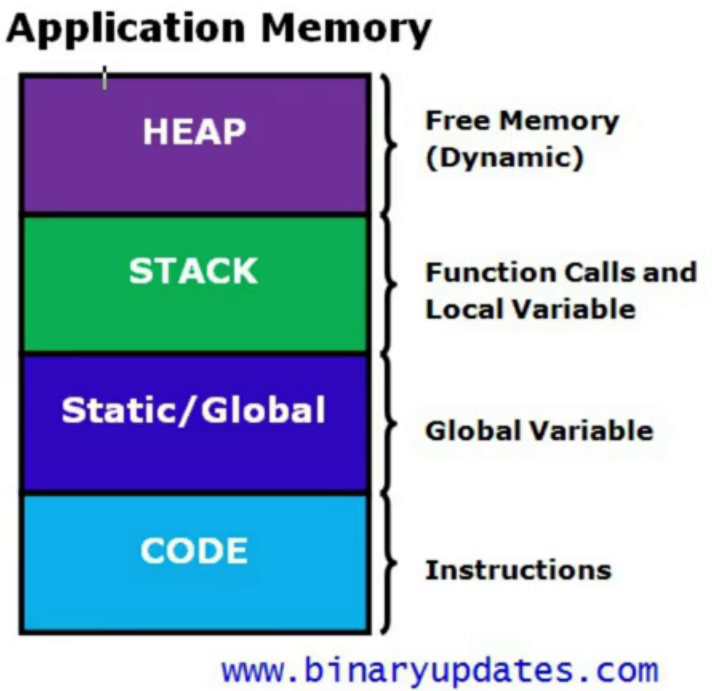
1-my recursion playlist (on my channel)

2-stack data structure

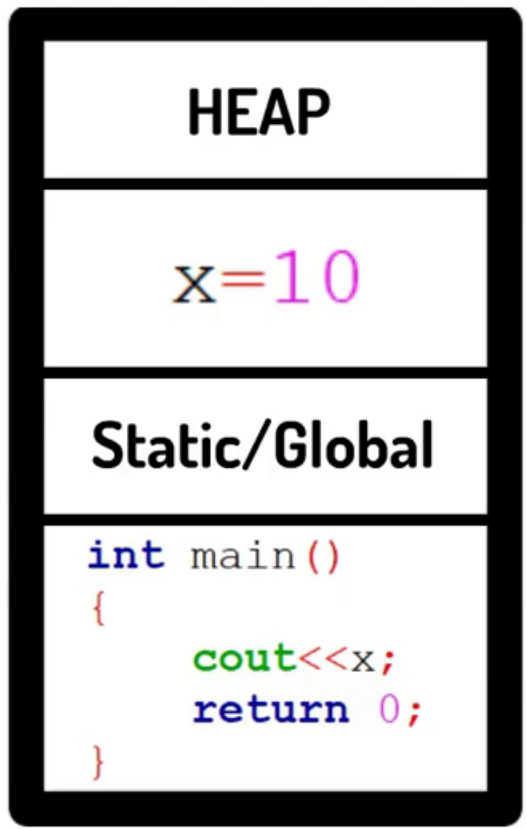
Recursion tree



Recursion memory




```
int main()
{
    int x=10;
    cout<<x;
    return 0;
}
```



Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Fibonacci

Global variables

=

=

Recursive function

python

```
def fn(n):
  if (n == 0 or n == 1):
    return n

  return fn(n-1) + fn(n-2)
```

Options

Enable step-by-step animation

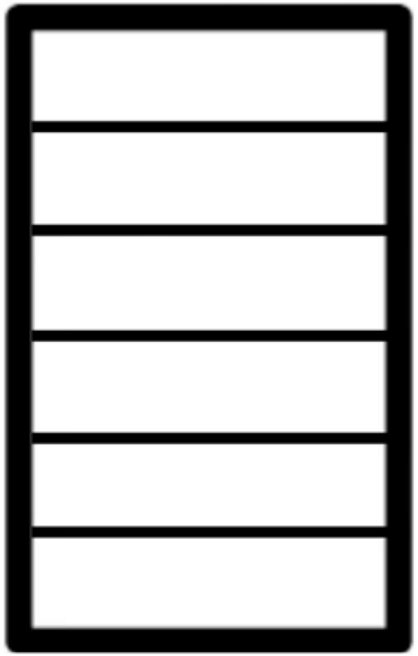
Enable memoization

Enable dark mode

fn(5) starts running

5

memory stack



fn(5)

Run

Made with ♥ by Bruno Papa • Github

rectangle-frame.png

Show all

Type here to search

92°F Clear

ENG

8:49 PM

8/11/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Fibonacci

Global variables

=

=

Recursive function python

```
def fn(n):
  if (n == 0 or n == 1):
    return n

  return fn(n-1) + fn(n-2)
```

Options

Enable step-by-step animation

Enable memoization

Enable dark mode

fn(5) Run

fn(5) starts running

5

memory stack

main()

Made with ♥ by Bruno Papa • Github

rectangle-frame.png

Show all

Type here to search

92°F Clear ENG 8:49 PM 8/11/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Fibonacci

Global variables

=

=

Recursive function python

```
def fn(n):
  if (n == 0 or n == 1):
    return n

  return fn(n-1) + fn(n-2)
```

Options

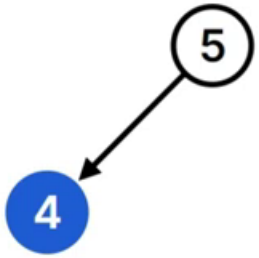
Enable step-by-step animation

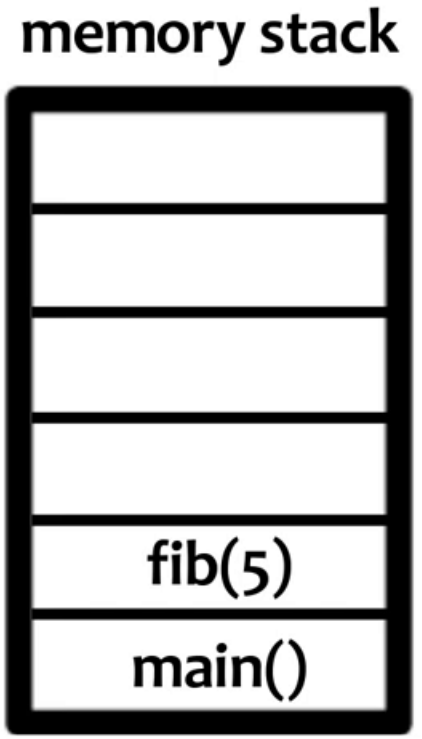
Enable memoization

Enable dark mode

fn(5) Run

fn(4) starts running





Made with ♥ by Bruno Papa • Github

rectangle-frame.png

Show all

Type here to search

92°F Clear ENG 8:50 PM 8/11/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Fibonacci

Global variables

=

=

Recursive function python

```
def fn(n):
  if (n == 0 or n == 1):
    return n

  return fn(n-1) + fn(n-2)
```

Options

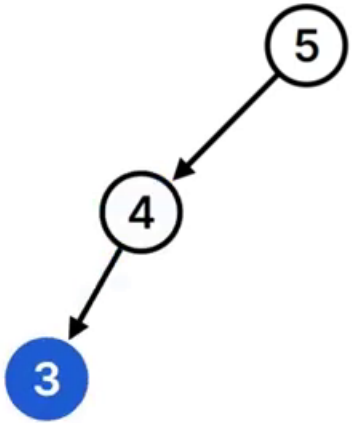
Enable step-by-step animation

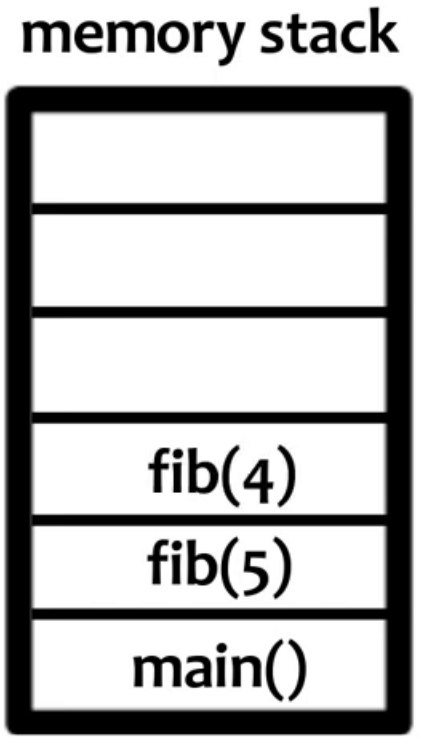
Enable memoization

Enable dark mode

fn(5) Run

fn(3) starts running





Made with ♥ by Bruno Papa • Github

rectangle-frame.png

Show all

Type here to search

92°F Clear ENG 8:50 PM 8/11/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Fibonacci

Global variables

=

=

Recursive function python

```
def fn(n):
  if (n == 0 or n == 1):
    return n

  return fn(n-1) + fn(n-2)
```

Options

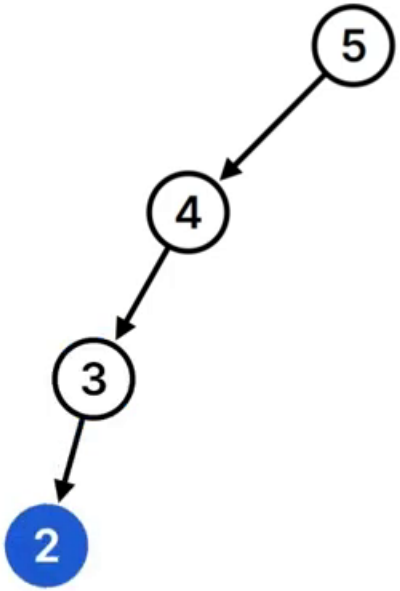
Enable step-by-step animation

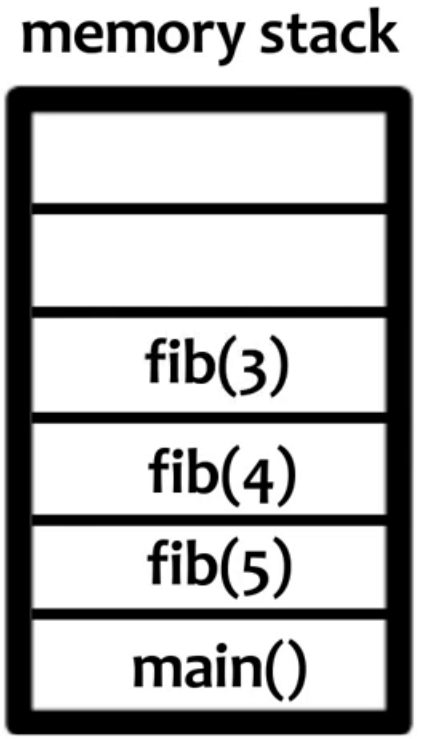
Enable memoization

Enable dark mode

fn(5) Run

fn(2) starts running





Made with ♥ by Bruno Papa • Github

rectangle-frame.png

Show all

Type here to search

92°F Clear ENG 8:50 PM 8/11/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Fibonacci

Global variables

=

=

Recursive function python

```
def fn(n):
  if (n == 0 or n == 1):
    return n

  return fn(n-1) + fn(n-2)
```

Options

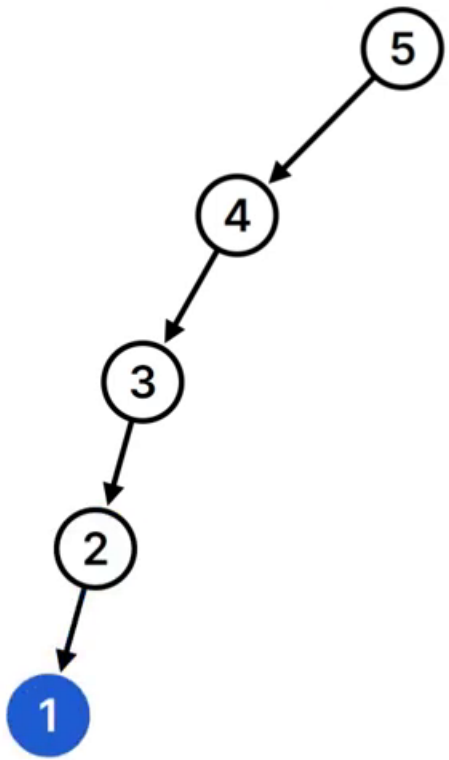
Enable step-by-step animation

Enable memoization

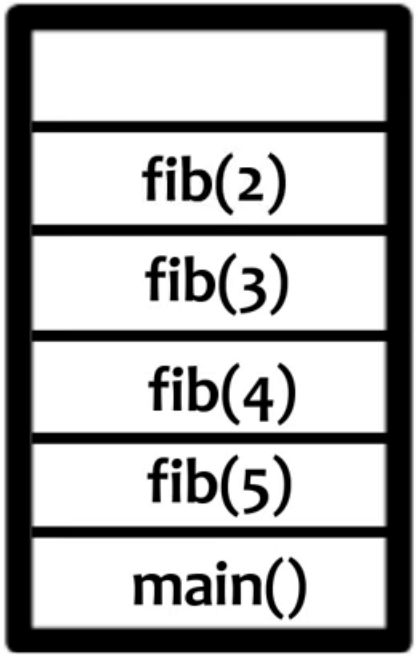
Enable dark mode

fn(5) Run

fn(1) starts running



memory stack



Made with ♥ by Bruno Papa • Github

rectangle-frame.png

Show all

Type here to search

92°F Clear ENG 8:50 PM 8/11/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Fibonacci

Global variables

=

=

Recursive function python

```
def fn(n):
  if (n == 0 or n == 1):
    return n

  return fn(n-1) + fn(n-2)
```

Options

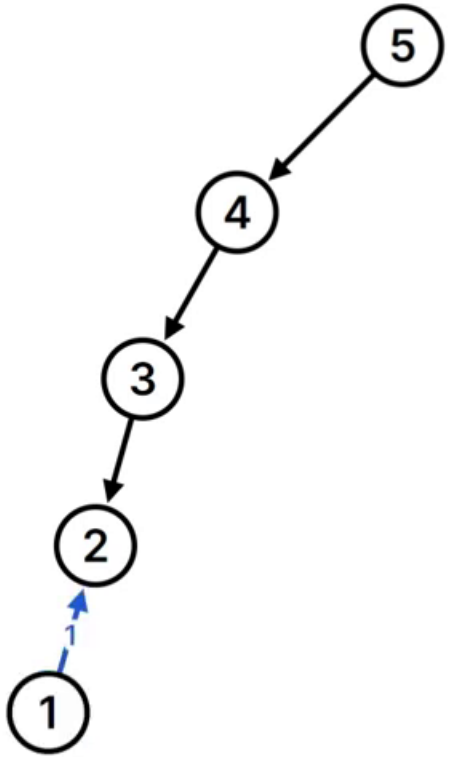
Enable step-by-step animation

Enable memoization

Enable dark mode

fn(5) Run

fn(1) returns 1 to fn(2)



memory stack

fib(1)

fib(2)

fib(3)

fib(4)

fib(5)

main()

Made with ♥ by Bruno Papa • Github

rectangle-frame.png

Show all

Type here to search

92°F Clear ENG 8:50 PM 8/11/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Fibonacci

Global variables

=

=

Recursive function python

```
def fn(n):
  if (n == 0 or n == 1):
    return n

  return fn(n-1) + fn(n-2)
```

Options

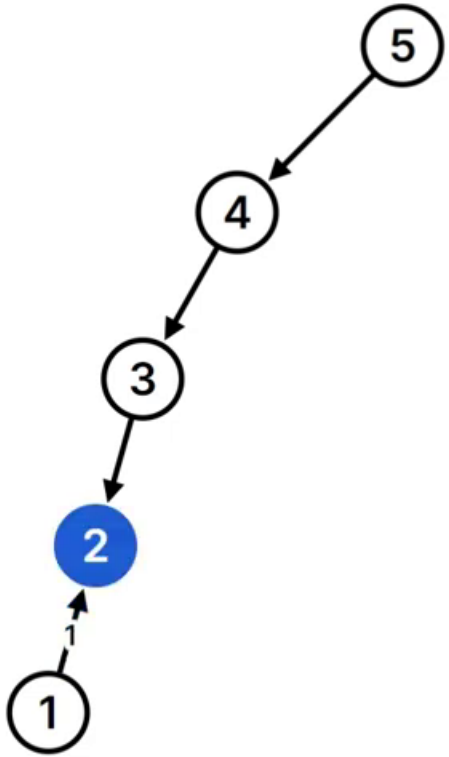
Enable step-by-step animation

Enable memoization

Enable dark mode

fn(5) Run

fn(2) continues running



memory stack

fib(2)

fib(3)

fib(4)

fib(5)

main()

Made with ♥ by Bruno Papa • Github

rectangle-frame.png

Show all

Type here to search

92°F Clear ENG 8:50 PM 8/11/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Fibonacci

Global variables

=

=

Recursive function python

```
def fn(n):
  if (n == 0 or n == 1):
    return n

  return fn(n-1) + fn(n-2)
```

Options

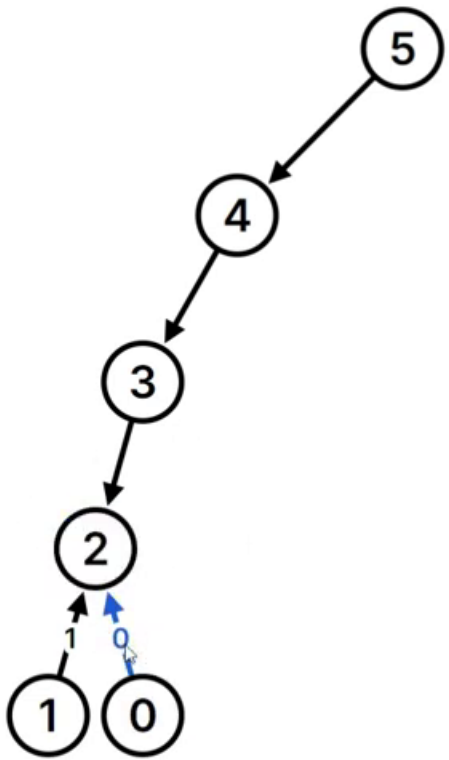
Enable step-by-step animation

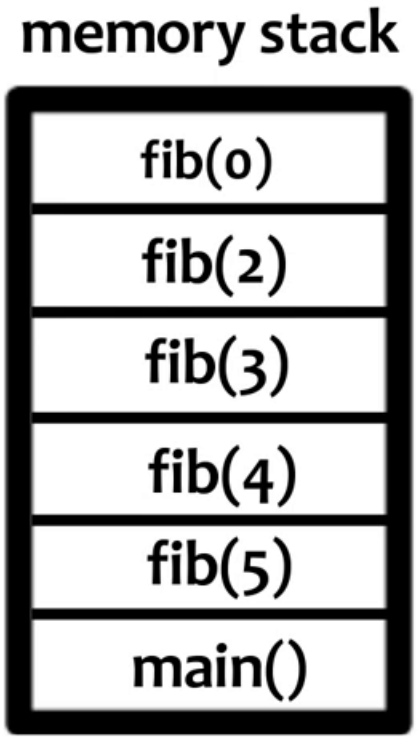
Enable memoization

Enable dark mode

fn(5) Run

fn(0) returns 0 to fn(2)





Made with ♥ by Bruno Papa • Github

rectangle-frame.png

Show all

Type here to search

92°F Clear ENG 8:50 PM 8/11/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Fibonacci

Global variables

=

=

Recursive function python

```
def fn(n):
  if (n == 0 or n == 1):
    return n

  return fn(n-1) + fn(n-2)
```

Options

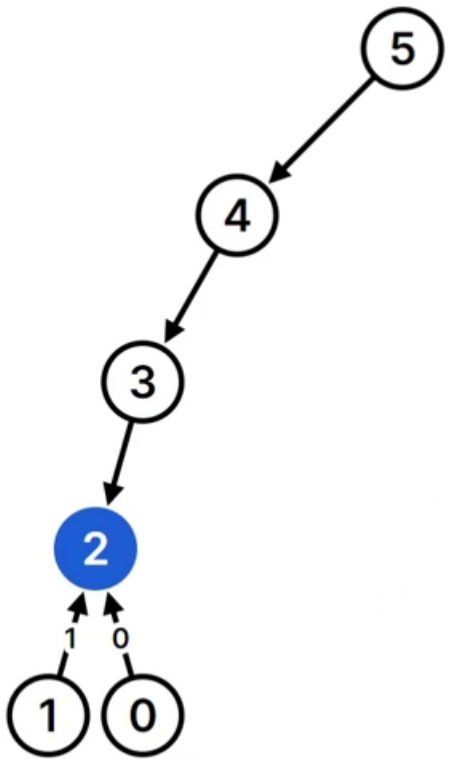
Enable step-by-step animation

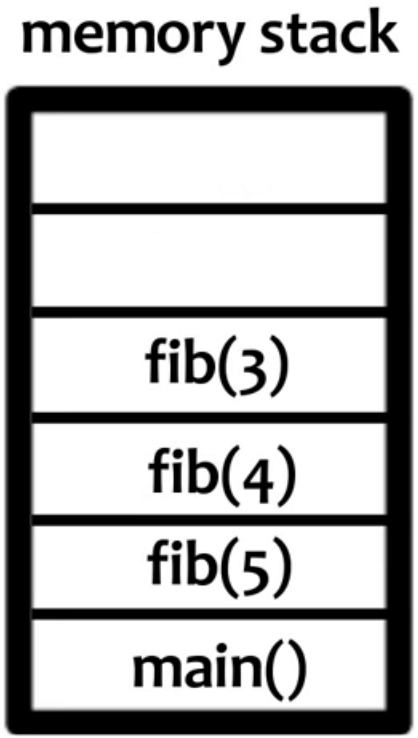
Enable memoization

Enable dark mode

fn(5) Run

fn(2) continues running





Made with ♥ by Bruno Papa • Github

rectangle-frame.png

Show all

Type here to search

92°F Clear ENG 8:51 PM 8/11/2021

Recursion Tree Visualizer

recursion.vercel.app

Pre-defined templates

Fibonacci

Global variables

=

=

Recursive function python

```
def fn(n):
  if (n == 0 or n == 1):
    return n

  return fn(n-1) + fn(n-2)
```

Options

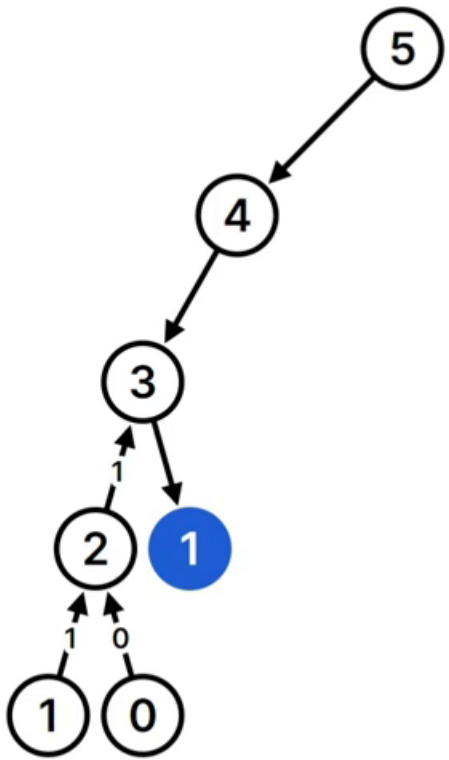
Enable step-by-step animation

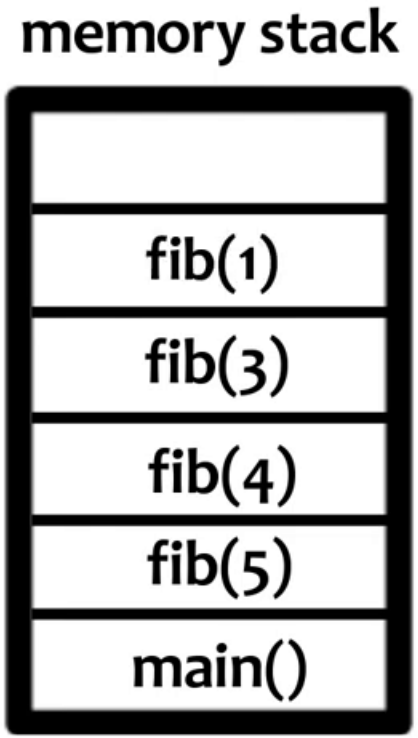
Enable memoization

Enable dark mode

fn(5) Run

fn(1) starts running





Made with ♥ by Bruno Papa • Github

rectangle-frame.png

Show all

Type here to search

92°F Clear ENG 8:51 PM 8/11/2021