

# Dynamic Programming and Overlapping Subproblems

Sérgio Oliveira Campos

Python Brasil 12

15 de outubro de 2016

# Sobre o Sérgio (seocam)

- Programador
- Pythonista
- Curte Open Source
- Algo mais que vale ser dito?



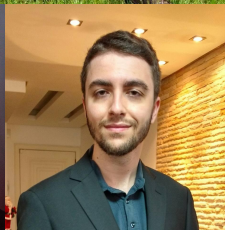
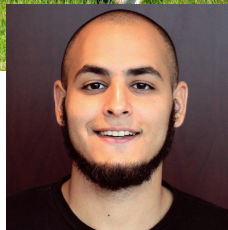
Crave Food Services



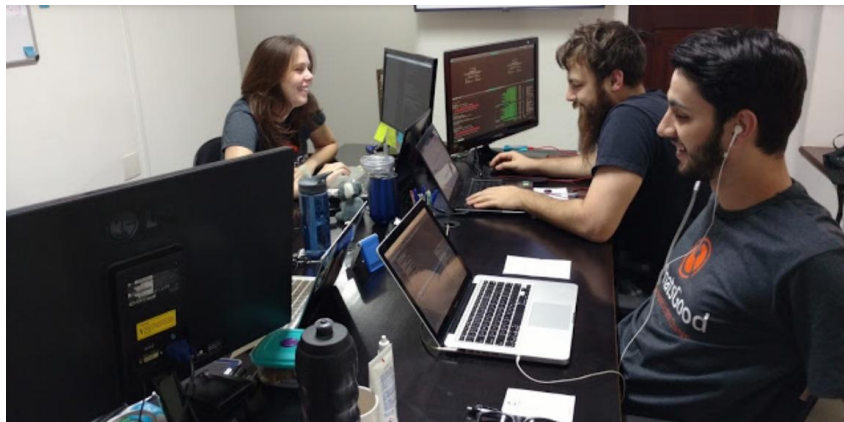
**WhatsGood**

Real Food System Change









Estamos contratando!

<http://crave.workable.com>

# Grupy-RP / Caipyra

[https://www.youtube.com/watch?v=YOT9i\\_4ILHs](https://www.youtube.com/watch?v=YOT9i_4ILHs)

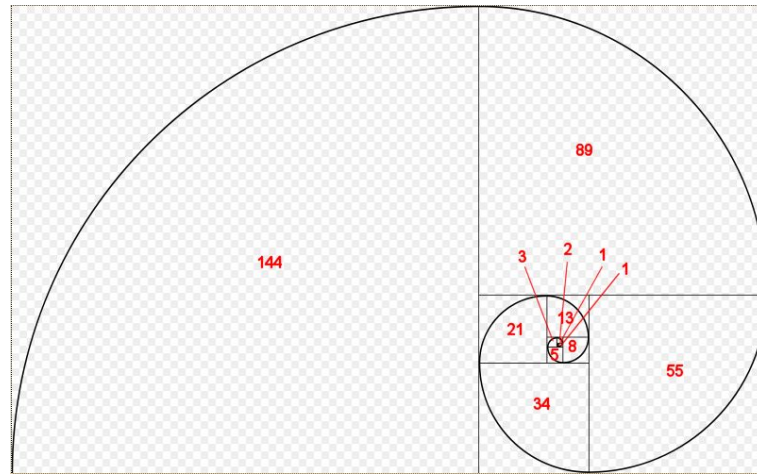


# Overlapping Subproblems

Problemas que podem ser divididos em sub-problemas e reusados consecutivamente até que a solução desejada seja obtida.

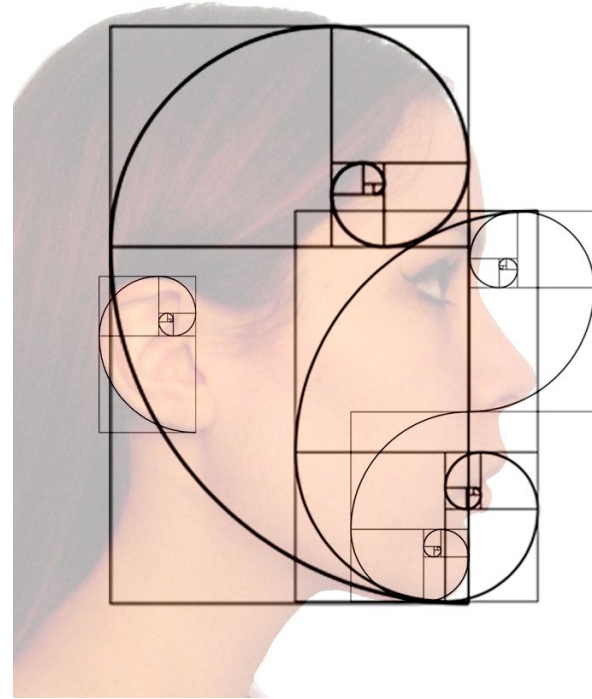
Para que um problema possa ser resolvido utilizando programação dinâmica o primeiro passo é encontrar a sua **relação de recorrência**.

# Fibonacci



**1 1 2 3 5 8 13 21 34 55 89 144**

# Fibonacci



# Fibonacci

## Implementação Recursiva

**fib1.py**



# Fibonacci

## Sobreposição de Subproblemas

```
f(5) = f(4) + f(3) = 5
|
|
|
|
|
|
|
f(4) = f(3) + f(2) = 3
|
|
|
f(3) = f(2) + f(1) = 2
|
|
|
f(2) = 1

|
|
|
|
|
|
|
f(3) = f(2) + f(1) = 2
|
|
|
f(2) = 1
|
|
|
f(1) = 1
```

# Fibonacci

## Implementação Iterativa

**fib2.py**

# Fibonacci

## Implementação Recursiva + Cache

**fib3.py**

Fibonacci

Implementação Recursiva + Cache + Decorator

**fib4.py**



# Fibonacci

## Implementação Recursiva + Functools

**fib5.py**

# Distância de Edição

o c - u r r - a n c e

o c c u r r e - n c e

0 substituição, 3 lacunas

o c u r r a n c e -

o c c u r r e n c e

6 substituições, 1 lacuna

o c - u r r a n c e

o c c u r r e n c e

1 substituição, 1 lacuna

# Distância de Edição

## Matriz de Memoização

		k	i	t	t	e	n
	0	1	2	3	4	5	6
s	1	1	2	3	4	5	6
i	2	2	1	2	3	4	5
t	3	3	2	1	2	3	4
t	4	4	3	2	1	2	3
i	5	5	4	3	2	2	3
n	6	6	5	4	3	3	2
g	7	7	6	5	4	4	3

		S	a	t	u	r	d	a	y
	0	1	2	3	4	5	6	7	8
S	1	0	1	2	3	4	5	6	7
u	2	1	1	2	2	3	4	5	6
n	3	2	2	2	3	3	4	5	6
d	4	3	3	3	3	4	3	4	5
a	5	4	3	4	4	4	4	3	4
y	6	5	4	4	5	5	5	4	3

# Distância de Edição

$$\text{lev}_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0, \\ \min \begin{cases} \text{lev}_{a,b}(i-1, j) + 1 \\ \text{lev}_{a,b}(i, j-1) + 1 \\ \text{lev}_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$



# Distância de Edição

## Implementação Recursiva

**`edit_distance.py`**

Distância de Edição

Implementação Recursiva + cache decorator

**`edit_distance2.py`**

# Distância de Edição

## Implementação Iterativa

**`edit_distance3.py`**

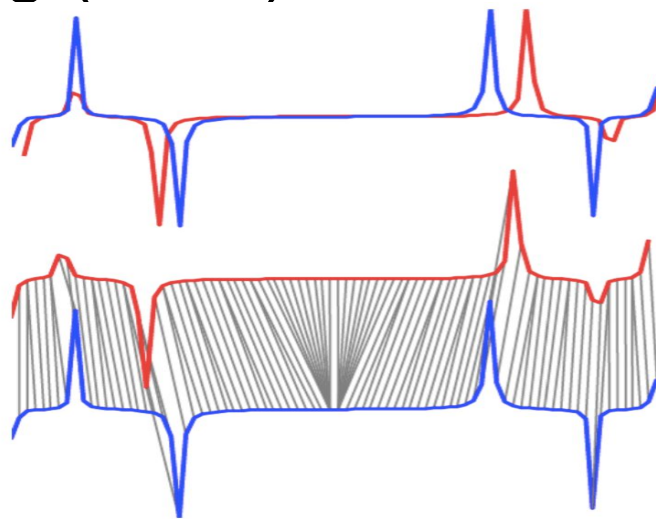
# Alinhamento de Sequências de DNA

Distância palavras **n** e **m** geralmente são  $< 10$

Sequências de DNA podem ter Gigabytes de  
informação

## Bônus: Dynamic Time Warping (DTW)

Usado para alinhamento de séries temporais.



Ver: [https://en.wikipedia.org/wiki/Dynamic\\_time\\_warping](https://en.wikipedia.org/wiki/Dynamic_time_warping)

# Mais a respeito do tema

<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00-introduction-to-computer-science-and-programming-fall-2008/video-lectures/lecture-13/>

<http://www.geeksforgeeks.org/dynamic-programming-set-5-edit-distance/>

<https://people.eecs.berkeley.edu/~vazirani/algorithms/chap6.pdf>

Como eu chego lá? Preparação

"Se mantenha curioso" – Aron Swartz

# Ajude a Treinar Jedis







# Desafios

*"The brick walls are not there to keep us out; the brick walls are there to give us a chance to show how badly we want something." – Randy Pausch*

# Obrigado!

Email: [seocam@seocam.com](mailto:seocam@seocam.com)

Telegram/Twitter/Github: @seocam