

CODING DOJO

MC102 - Algoritmos e
Programação de
Computadores

Santiago Valdés Ravelo
[https://ic.unicamp.br/~santiago/
ravelo@unicamp.br](https://ic.unicamp.br/~santiago/ravelo@unicamp.br)

04/25

8



UNICAMP



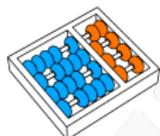


“O dojo é um local de superação, vínculo familiar e propagação da sabedoria ninja.”

Sensei Cícero Melo Hosho Ryu Ninpo.



DÚVIDAS DA AULA ANTERIOR

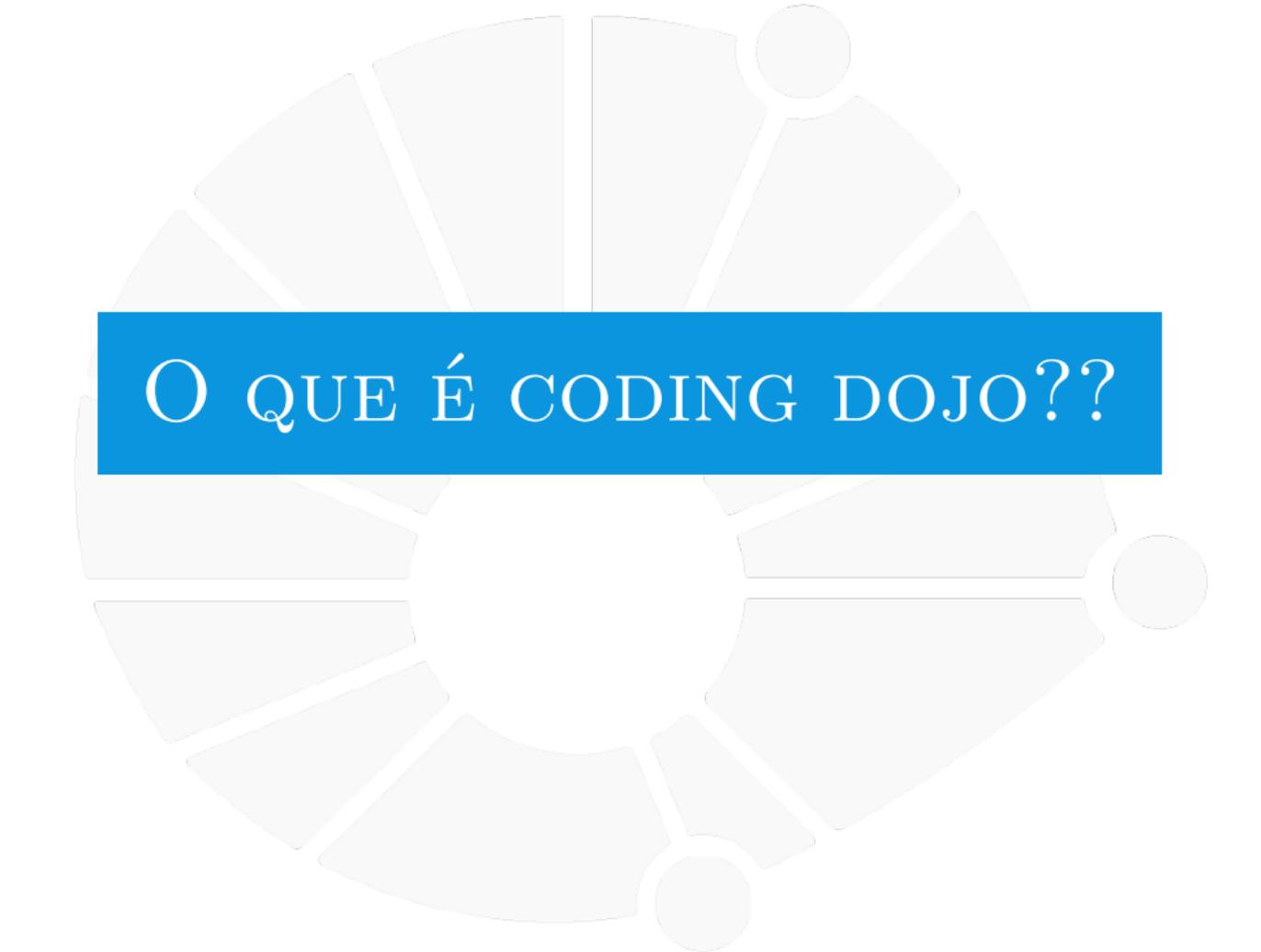


Dúvidas selecionadas

- ▶ Não entendi a função `lista.index()`.
- ▶ Não entendi como funciona o `slice`.
- ▶ Não entendi o uso das funções `list.pop()` e `list.append()`.
- ▶ Como faço para criar uma lista com a quantidade de elementos que quero, sem usar `append` varias vezes?
- ▶ Como funciona mesmo o `for` dentro de `for`?
- ▶ Não entendi muito bem a diferença entre `break` e `continue`.
- ▶ Não entendi como usar o `"range"` fora da estrutura do `for`.
- ▶ Se uma lista `L` contém um número inteiro `n` escrito como `float` (com casa decimal igual a zero), e `m` é o mesmo inteiro escrito em `int`, a expressão `"m in L"` é `True` ou `False`? Por exemplo, se a lista contém o `10.0` (`float`), a expressão `"10 in L"` será `True`, mesmo que os tipos sejam diferentes?
- ▶ É possível atribuir diferentes valores pra diferentes posições de uma lista de modo padronizado? Por exemplo, fazer todo número que esteja numa posição ímpar da lista receber o valor de seu dobro.
- ▶ Como faço para somar todos os índices de uma lista com outra lista?
- ▶ É possível subtrair uma lista de outra? Exemplo `L1 = [1,2,3,4]` e `L2 = [1,2]`. `L1-L2 = [3,4]`?
- ▶ É mais rápido usar as funções nativas do python ou fazer manualmente as operações com listas?
- ▶ Não entendi muito bem a solução revisitada do Crivo.



CODING
DOJO



O QUE É CODING DOJO??



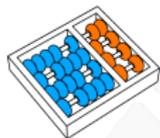
Coding dojo

Inspirado nas práticas dos dojos de artes marciais japonesas, um Coding Dojo pode ter vários estilos.

O formato que usaremos será o: Randori Kata.

Componentes:

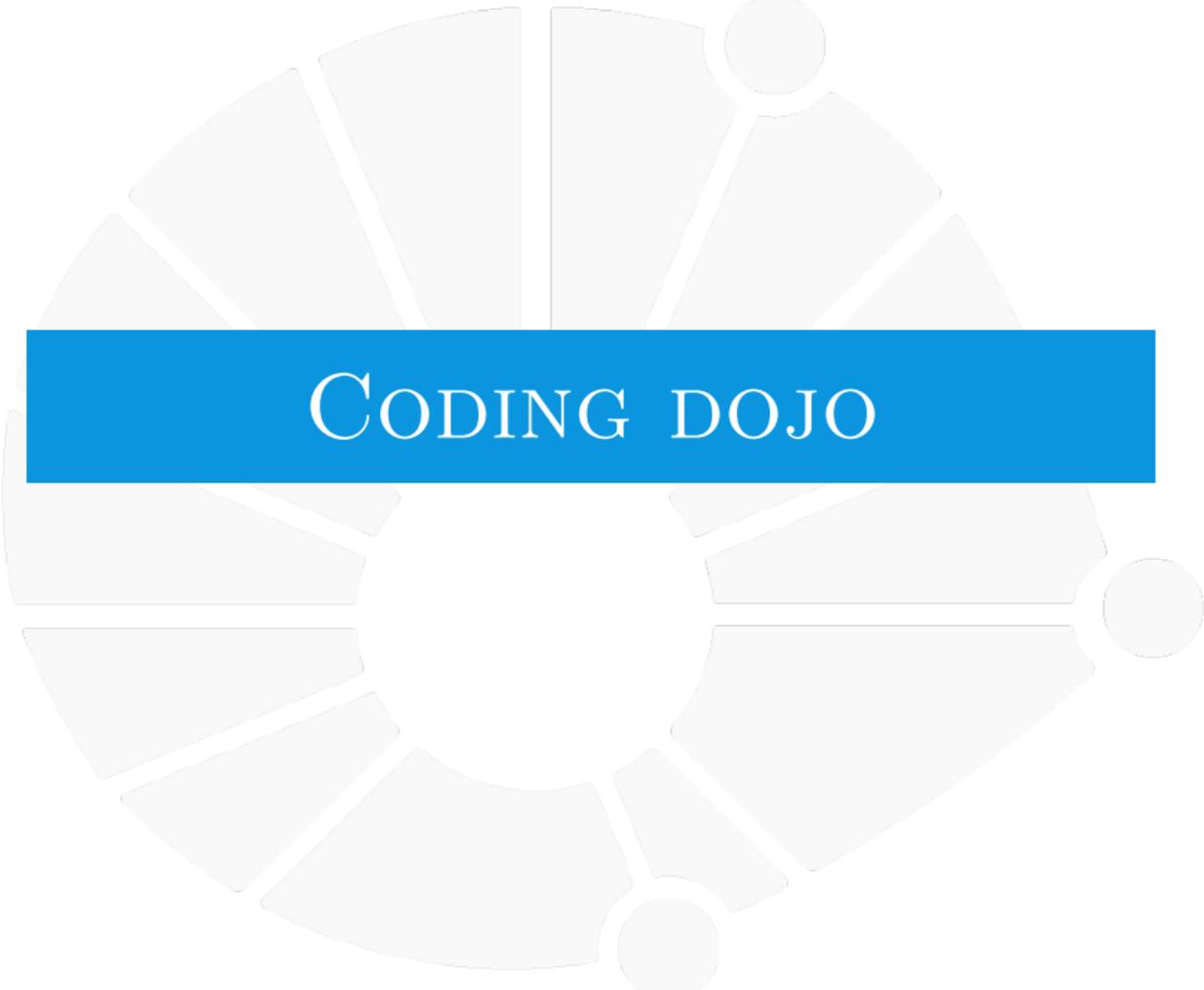
- ▶ Um computador.
- ▶ Um projetor.
- ▶ Um mestre (o tal de Santiago?).
- ▶ Um piloto (aluno voluntário).
- ▶ Um copiloto (aluno voluntário).
- ▶ Plateia (restante da turma).



Coding dojo - Randori Kata

Funcionamento:

- ▶ O mestre propõe um problema.
- ▶ O piloto e o copiloto tentam solucioná-lo durante 5 minutos:
 - ▶ Piloto e copiloto devem explicar a ideia de solução para a plateia.
 - ▶ Só o piloto pode programar.
 - ▶ O copiloto pode apontar erros e dar sugestões.
 - ▶ A plateia só pode participar se o piloto ou o copiloto pedem ajuda.
 - ▶ Em caso de necessidade, podem perguntar ao mestre, mas ele responde com outra pergunta.
- ▶ Passados os 5 minutos, o desafio pausa (mesmo não sendo solucionado):
 - ▶ O piloto volta para a plateia.
 - ▶ O copiloto se torna piloto.
 - ▶ Um novo membro da plateia se torna copiloto.
- ▶ A solução do desafio continua com os novos piloto e copiloto.
- ▶ Se o desafio for concluído, o mestre lança um novo desafio.

The logo features a large, light gray circular shape composed of several wedge-shaped segments, resembling a fan or a stylized 'C'. Three small gray circles are positioned around the perimeter of this shape. A solid blue horizontal bar is centered across the middle of the circular graphic.

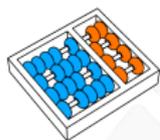
CODING DOJO



Prontos para começar?



**CODE
DOJO**



Mínimos locais

Dada uma sequência de números reais S , um valor é um mínimo local, se os dois adjacentes a ele forem maiores (isto é, $s_{i-1} > s_i < s_{i+1}$). Faça um programa que leia uma sequência de inteiros S e imprima as posições dos mínimos locais nela.

Exemplo de entrada:

8 2 1 2 5 2 4 7 5 2 6 2 4

Exemplo de saída (para a entrada do exemplo):

3 6 10 12



Menor, maior, média, mediana e moda

Faça um programa que leia uma sequência de inteiros S e imprima o menor valor da sequência, o maior, a média (até duas casas decimais), a mediana (elemento com igual número de maiores (ou iguais) que de menores ou iguais) e a moda (elemento que mais se repete).

Exemplo de entrada:

8 2 1 2 5 2 4 7 5 2 6 2 4

Exemplo de saída (para a entrada do exemplo):

Menor: 1

Maior: 8

Media: 3.84

Mediana: 4

Moda: 2



Subsequência

Uma sequência S_1 é subsequência de uma S_2 se todos os elementos de S_1 aparecem de forma consecutiva e na mesma ordem em S_2 . Faça um programa que leia duas sequências de inteiros S_1 e S_2 e imprima se S_1 é ou não subsequência de S_2 .

Exemplo de entrada:

```
2 6 2 4
8 2 1 2 6 2 4 7 5
```

Exemplo de saída (para a entrada do exemplo):

[2, 6, 2, 4] eh subsequencia de [8, 2, 1, 2, 6, 2, 4, 7, 5].



União de vetores ordenados

Faça um programa que:

- Leia duas sequências de números reais S_1 e outra S_2 . Assumindo que os elementos de S_1 estejam ordenados e que os de S_2 também estejam ordenados, imprima (em ordem) os elementos da união entre S_1 e S_2 .
- Solucione o item anterior sem usar `sort()`.

Exemplo de entrada:

```
2 4 7
1 5 6 8
```

Exemplo de saída (para a entrada do exemplo):

```
1 2 4 5 6 7 8
```



Subsequência de soma máxima

Faça um programa que leia uma sequência de reais S e encontrar duas posições i e j na sequência, tais que a soma dos elementos desde a posição i até a j seja máxima ($i \leq j$). Imprima as posições e o valor da soma dos elementos da posição i até a j .

Exemplo de entrada:

3 4 - 5 3 3 - 8 2

Exemplo de saída (para a entrada do exemplo):

Intervalo: [1, 5]

Soma: 8



Três somam zero?

Faça um programa que leia uma sequência de reais S e, se houver três valores na sequência cuja soma é zero, os imprima, caso contrário imprima que não há três valores cuja soma é zero.

Exemplo de entrada:

3 4 -5 3 3 -8 2

Exemplo de saída (para a entrada do exemplo):

3 + -5 + 2 = 0



Escolhendo números

Este é um jogo entre duas pessoas A e B que se alternam escolhendo números em uma dada sequência S com quantidade par de elementos. As regras são as seguintes:

- ▶ Começa o jogador A .
- ▶ Na sua vez, o jogador escolhe um dos extremos da sequência S e o remove da sequência, passando o turno para o outro jogador.
- ▶ O jogo acaba quando todos os números foram removidos.
- ▶ Ganha o jogador que removeu a maior soma.

Sabendo que você é o jogador A e supondo que o B é um especialista no jogo. Crie uma estratégia em que A nunca perde! Faça um programa que leia a sequência de números e imprima a soma dos números selecionados por A seguindo sua estratégia.

Exemplo de entrada:

10 1 1 1 1 1

Exemplo de saída (para a entrada do exemplo):

12

CODING DOJO

MC102 - Algoritmos e
Programação de
Computadores

Santiago Valdés Ravelo
[https://ic.unicamp.br/~santiago/
ravelo@unicamp.br](https://ic.unicamp.br/~santiago/ravelo@unicamp.br)

04/25

8



UNICAMP

