

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)

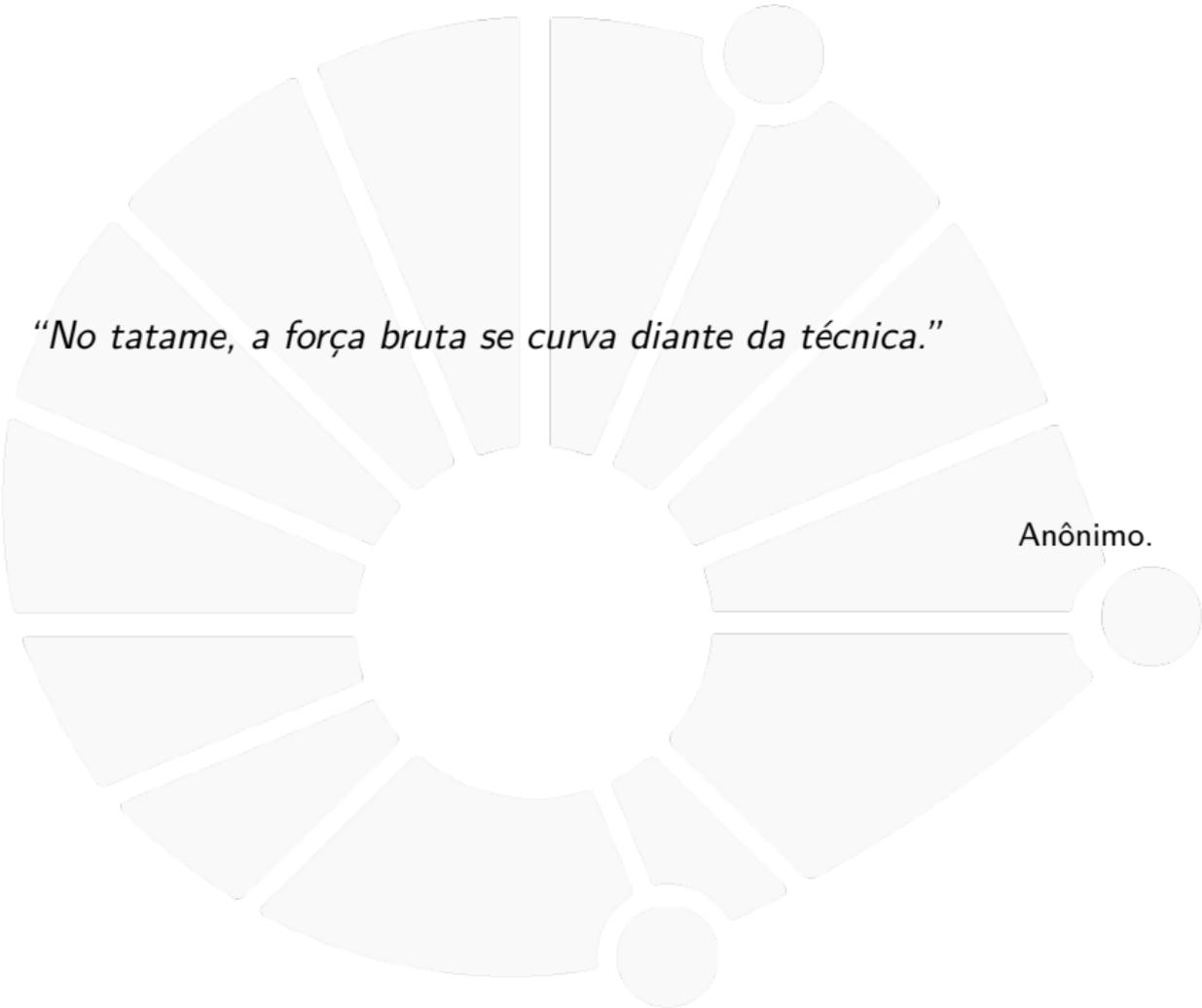
05/25

17



UNICAMP



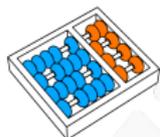


“No tatame, a força bruta se curva diante da técnica.”

Anônimo.

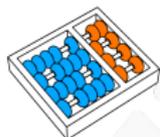


DÚVIDAS DA AULA ANTERIOR



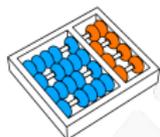
Dúvidas selecionadas

- ▶ Eu entendo a praticidade de colocar "tudo" numa linha, mas isso também perde, na minha visão, muito da estética do código. Alguns programadores optam por não fazer dessa forma por essa ou outras razões?



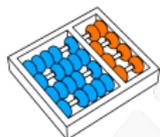
Dúvidas selecionadas

- ▶ Eu entendo a praticidade de colocar "tudo" numa linha, mas isso também perde, na minha visão, muito da estética do código. Alguns programadores optam por não fazer dessa forma por essa ou outras razões?
- ▶ Como eu sei se deveria usar uma tupla ou uma lista?



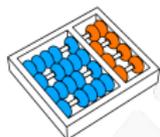
Dúvidas selecionadas

- ▶ Eu entendo a praticidade de colocar "tudo" numa linha, mas isso também perde, na minha visão, muito da estética do código. Alguns programadores optam por não fazer dessa forma por essa ou outras razões?
- ▶ Como eu sei se deveria usar uma tupla ou uma lista?
- ▶ Não entendi como funciona o if in-line sozinho sem ser dentro de uma lista.



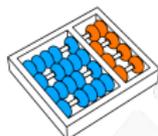
Dúvidas selecionadas

- ▶ Eu entendo a praticidade de colocar "tudo" numa linha, mas isso também perde, na minha visão, muito da estética do código. Alguns programadores optam por não fazer dessa forma por essa ou outras razões?
- ▶ Como eu sei se deveria usar uma tupla ou uma lista?
- ▶ Não entendi como funciona o if in-line sozinho sem ser dentro de uma lista.
- ▶ Porque hash não é ordenável?



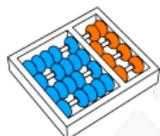
Dúvidas selecionadas

- ▶ Eu entendo a praticidade de colocar "tudo" numa linha, mas isso também perde, na minha visão, muito da estética do código. Alguns programadores optam por não fazer dessa forma por essa ou outras razões?
- ▶ Como eu sei se deveria usar uma tupla ou uma lista?
- ▶ Não entendi como funciona o if in-line sozinho sem ser dentro de uma lista.
- ▶ Porque hash não é ordenável?
- ▶ Não compreendi as tuplas, elas são a mesma coisa que uma variável? Ou uma lista? (Ou ambos)



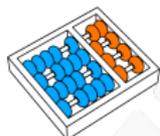
Dúvidas selecionadas

- ▶ Eu entendo a praticidade de colocar "tudo" numa linha, mas isso também perde, na minha visão, muito da estética do código. Alguns programadores optam por não fazer dessa forma por essa ou outras razões?
- ▶ Como eu sei se deveria usar uma tupla ou uma lista?
- ▶ Não entendi como funciona o if in-line sozinho sem ser dentro de uma lista.
- ▶ Porque hash não é ordenável?
- ▶ Não compreendi as tuplas, elas são a mesma coisa que uma variável? Ou uma lista? (Ou ambos)
- ▶ É possível fazer matrizes e outras estruturas n-dimensionais usando tuplas?



Dúvidas selecionadas

- ▶ Eu entendo a praticidade de colocar "tudo" numa linha, mas isso também perde, na minha visão, muito da estética do código. Alguns programadores optam por não fazer dessa forma por essa ou outras razões?
- ▶ Como eu sei se deveria usar uma tupla ou uma lista?
- ▶ Não entendi como funciona o if in-line sozinho sem ser dentro de uma lista.
- ▶ Porque hash não é ordenável?
- ▶ Não compreendi as tuplas, elas são a mesma coisa que uma variável? Ou uma lista? (Ou ambos)
- ▶ É possível fazer matrizes e outras estruturas n-dimensionais usando tuplas?
- ▶ Como eu faço para acessar um dicionário que está dentro de um dicionário?



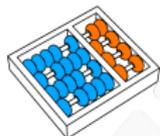
Dúvidas selecionadas

- ▶ Eu entendo a praticidade de colocar "tudo" numa linha, mas isso também perde, na minha visão, muito da estética do código. Alguns programadores optam por não fazer dessa forma por essa ou outras razões?
- ▶ Como eu sei se deveria usar uma tupla ou uma lista?
- ▶ Não entendi como funciona o if in-line sozinho sem ser dentro de uma lista.
- ▶ Porque hash não é ordenável?
- ▶ Não compreendi as tuplas, elas são a mesma coisa que uma variável? Ou uma lista? (Ou ambos)
- ▶ É possível fazer matrizes e outras estruturas n-dimensionais usando tuplas?
- ▶ Como eu faço para acessar um dicionário que está dentro de um dicionário?
- ▶ Consigo criar novos tipos de objetos que sejam imutáveis?



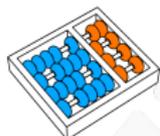
Dúvidas selecionadas

- ▶ Eu entendo a praticidade de colocar "tudo" numa linha, mas isso também perde, na minha visão, muito da estética do código. Alguns programadores optam por não fazer dessa forma por essa ou outras razões?
- ▶ Como eu sei se deveria usar uma tupla ou uma lista?
- ▶ Não entendi como funciona o if in-line sozinho sem ser dentro de uma lista.
- ▶ Porque hash não é ordenável?
- ▶ Não compreendi as tuplas, elas são a mesma coisa que uma variável? Ou uma lista? (Ou ambos)
- ▶ É possível fazer matrizes e outras estruturas n-dimensionais usando tuplas?
- ▶ Como eu faço para acessar um dicionário que está dentro de um dicionário?
- ▶ Consigo criar novos tipos de objetos que sejam imutáveis?
- ▶ Não entendi muito bem como faz uma matriz com list comprehension.



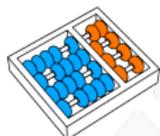
Dúvidas selecionadas

- ▶ Eu entendo a praticidade de colocar "tudo" numa linha, mas isso também perde, na minha visão, muito da estética do código. Alguns programadores optam por não fazer dessa forma por essa ou outras razões?
- ▶ Como eu sei se deveria usar uma tupla ou uma lista?
- ▶ Não entendi como funciona o if in-line sozinho sem ser dentro de uma lista.
- ▶ Porque hash não é ordenável?
- ▶ Não compreendi as tuplas, elas são a mesma coisa que uma variável? Ou uma lista? (Ou ambos)
- ▶ É possível fazer matrizes e outras estruturas n-dimensionais usando tuplas?
- ▶ Como eu faço para acessar um dicionário que está dentro de um dicionário?
- ▶ Consigo criar novos tipos de objetos que sejam imutáveis?
- ▶ Não entendi muito bem como faz uma matriz com list comprehension.
- ▶ map retorna um tuple?



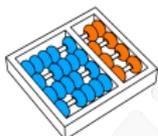
Dúvidas selecionadas

- ▶ Eu entendo a praticidade de colocar "tudo" numa linha, mas isso também perde, na minha visão, muito da estética do código. Alguns programadores optam por não fazer dessa forma por essa ou outras razões?
- ▶ Como eu sei se deveria usar uma tupla ou uma lista?
- ▶ Não entendi como funciona o if in-line sozinho sem ser dentro de uma lista.
- ▶ Porque hash não é ordenável?
- ▶ Não compreendi as tuplas, elas são a mesma coisa que uma variável? Ou uma lista? (Ou ambos)
- ▶ É possível fazer matrizes e outras estruturas n-dimensionais usando tuplas?
- ▶ Como eu faço para acessar um dicionário que está dentro de um dicionário?
- ▶ Consigo criar novos tipos de objetos que sejam imutáveis?
- ▶ Não entendi muito bem como faz uma matriz com list comprehension.
- ▶ map retorna um tuple?
- ▶ Posso usar not, or e and na compreensão de lista?



Dúvidas selecionadas

- ▶ Eu entendo a praticidade de colocar "tudo" numa linha, mas isso também perde, na minha visão, muito da estética do código. Alguns programadores optam por não fazer dessa forma por essa ou outras razões?
- ▶ Como eu sei se deveria usar uma tupla ou uma lista?
- ▶ Não entendi como funciona o if in-line sozinho sem ser dentro de uma lista.
- ▶ Porque hash não é ordenável?
- ▶ Não compreendi as tuplas, elas são a mesma coisa que uma variável? Ou uma lista? (Ou ambos)
- ▶ É possível fazer matrizes e outras estruturas n-dimensionais usando tuplas?
- ▶ Como eu faço para acessar um dicionário que está dentro de um dicionário?
- ▶ Consigo criar novos tipos de objetos que sejam imutáveis?
- ▶ Não entendi muito bem como faz uma matriz com list comprehension.
- ▶ map retorna um tuple?
- ▶ Posso usar not, or e and na compreensão de lista?
- ▶ Por que usar as list comprehensions é mais rápido computacionalmente do que iterar por um range?



Dúvidas selecionadas

- ▶ Eu entendo a praticidade de colocar "tudo" numa linha, mas isso também perde, na minha visão, muito da estética do código. Alguns programadores optam por não fazer dessa forma por essa ou outras razões?
- ▶ Como eu sei se deveria usar uma tupla ou uma lista?
- ▶ Não entendi como funciona o if in-line sozinho sem ser dentro de uma lista.
- ▶ Porque hash não é ordenável?
- ▶ Não compreendi as tuplas, elas são a mesma coisa que uma variável? Ou uma lista? (Ou ambos)
- ▶ É possível fazer matrizes e outras estruturas n-dimensionais usando tuplas?
- ▶ Como eu faço para acessar um dicionário que está dentro de um dicionário?
- ▶ Consigo criar novos tipos de objetos que sejam imutáveis?
- ▶ Não entendi muito bem como faz uma matriz com list comprehension.
- ▶ map retorna um tuple?
- ▶ Posso usar not, or e and na compreensão de lista?
- ▶ Por que usar as list comprehensions é mais rápido computacionalmente do que iterar por um range?
- ▶ Por que é possível fazer uma tupla de listas mas não é possível usar a tupla de listas como chave para um dicionário?



CODING
DOJO

The background features a stylized, light gray fan with several segments radiating from a central point. Two small circles are positioned at the top and bottom of the fan's structure. A solid blue horizontal rectangle is centered over the fan, containing the title text in white.

LEBRANDO O FUNCIONAMENTO DO CODING DOJO



Coding Dojo

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



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.



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:



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.



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.



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



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?).



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



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).



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



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).



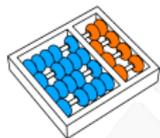
Coding Dojo

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



Coding Dojo

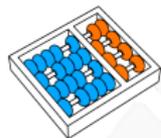
Inspirado nas práticas dos dojos de artes marciais japonesas, um Coding Dojo pode ter vários estilos.

O formato que usaremos será o: Randori Kata.

Componentes:

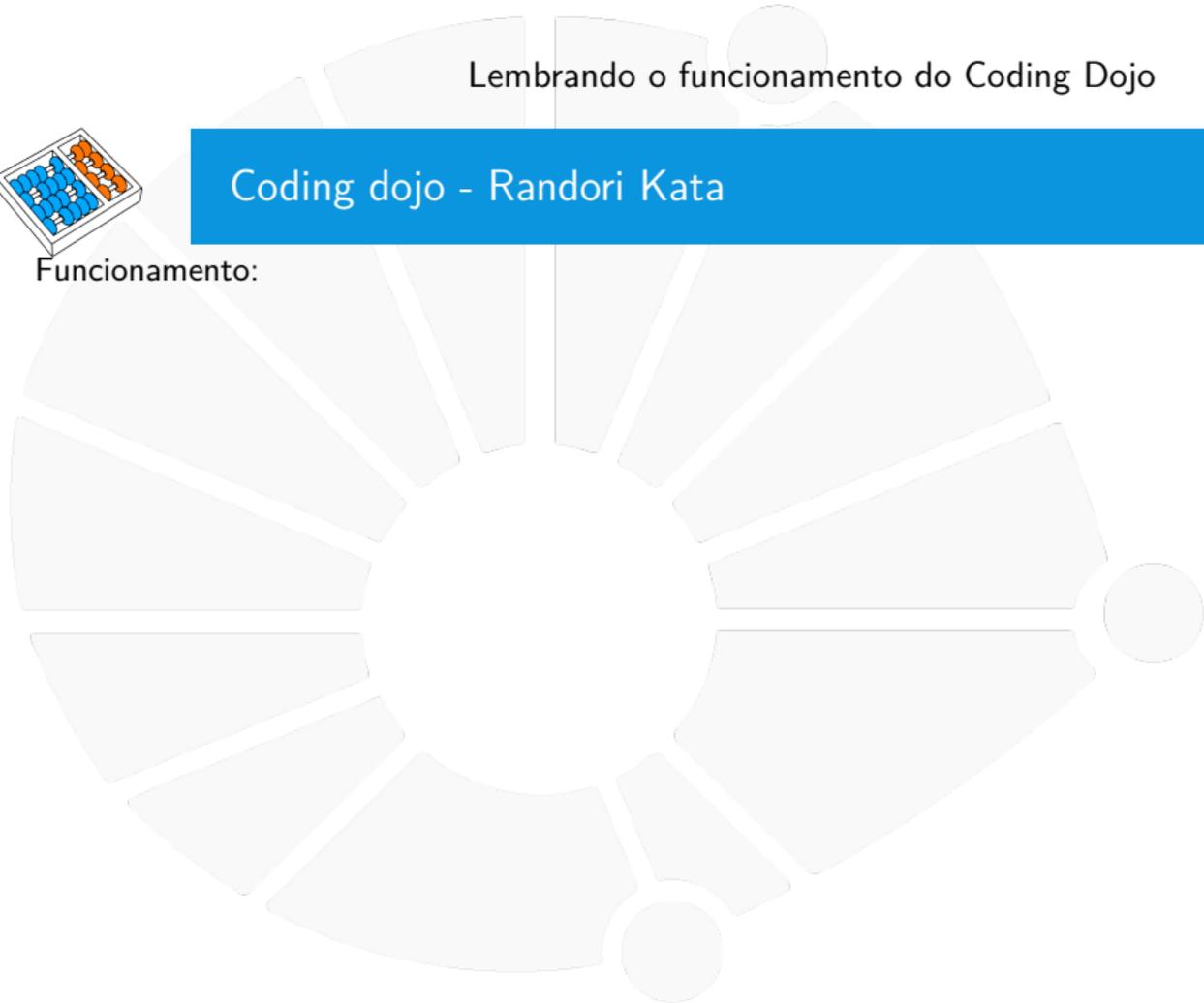
- ▶ Um computador.
- ▶ Um projetor.
- ▶ Um mestre (o tal de Santiago?).
- ▶ Um piloto (aluno voluntário).
- ▶ Um copiloto (aluno voluntário).
- ▶ Plateia (restante da turma).

Lembrando o funcionamento do Coding Dojo



Coding dojo - Randori Kata

Funcionamento:

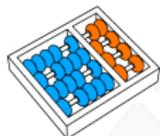




Coding dojo - Randori Kata

Funcionamento:

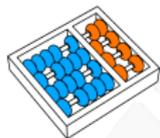
- ▶ O mestre propõe um problema.



Coding dojo - Randori Kata

Funcionamento:

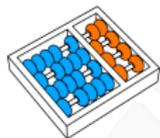
- ▶ O mestre propõe um problema.
- ▶ O piloto e o copiloto tentam solucioná-lo durante 5 minutos:



Coding dojo - Randori Kata

Funcionamento:

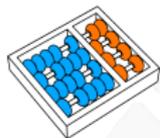
- ▶ O mestre propõe um problema.
- ▶ O piloto e o copiloto tentam solucioná-lo durante 5 minutos:
 - ▶ Piloto e copiloto devem explicar a ideia de solução para a plateia.



Coding dojo - Randori Kata

Funcionamento:

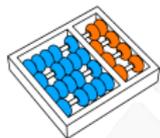
- ▶ O mestre propõe um problema.
- ▶ O piloto e o copiloto tentam solucioná-lo durante 5 minutos:
 - ▶ Piloto e copiloto devem explicar a ideia de solução para a plateia.
 - ▶ Só o piloto pode programar.



Coding dojo - Randori Kata

Funcionamento:

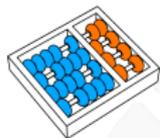
- ▶ O mestre propõe um problema.
- ▶ O piloto e o copiloto tentam solucioná-lo durante 5 minutos:
 - ▶ Piloto e copiloto devem explicar a ideia de solução para a plateia.
 - ▶ Só o piloto pode programar.
 - ▶ O copiloto pode apontar erros e dar sugestões.



Coding dojo - Randori Kata

Funcionamento:

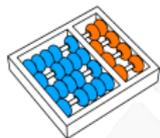
- ▶ O mestre propõe um problema.
- ▶ O piloto e o copiloto tentam solucioná-lo durante 5 minutos:
 - ▶ Piloto e copiloto devem explicar a ideia de solução para a plateia.
 - ▶ Só o piloto pode programar.
 - ▶ O copiloto pode apontar erros e dar sugestões.
 - ▶ A plateia só pode participar se o piloto ou o copiloto pedem ajuda.



Coding dojo - Randori Kata

Funcionamento:

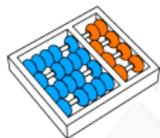
- ▶ O mestre propõe um problema.
- ▶ O piloto e o copiloto tentam solucioná-lo durante 5 minutos:
 - ▶ Piloto e copiloto devem explicar a ideia de solução para a plateia.
 - ▶ Só o piloto pode programar.
 - ▶ O copiloto pode apontar erros e dar sugestões.
 - ▶ A plateia só pode participar se o piloto ou o copiloto pedem ajuda.
 - ▶ Em caso de necessidade, podem perguntar ao mestre, mas ele responde com outra pergunta.



Coding dojo - Randori Kata

Funcionamento:

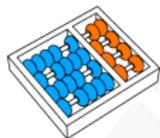
- ▶ O mestre propõe um problema.
- ▶ O piloto e o copiloto tentam solucioná-lo durante 5 minutos:
 - ▶ Piloto e copiloto devem explicar a ideia de solução para a plateia.
 - ▶ Só o piloto pode programar.
 - ▶ O copiloto pode apontar erros e dar sugestões.
 - ▶ A plateia só pode participar se o piloto 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):



Coding dojo - Randori Kata

Funcionamento:

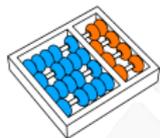
- ▶ O mestre propõe um problema.
- ▶ O piloto e o copiloto tentam solucioná-lo durante 5 minutos:
 - ▶ Piloto e copiloto devem explicar a ideia de solução para a plateia.
 - ▶ Só o piloto pode programar.
 - ▶ O copiloto pode apontar erros e dar sugestões.
 - ▶ A plateia só pode participar se o piloto 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.



Coding dojo - Randori Kata

Funcionamento:

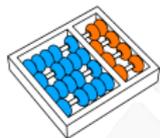
- ▶ O mestre propõe um problema.
- ▶ O piloto e o copiloto tentam solucioná-lo durante 5 minutos:
 - ▶ Piloto e copiloto devem explicar a ideia de solução para a plateia.
 - ▶ Só o piloto pode programar.
 - ▶ O copiloto pode apontar erros e dar sugestões.
 - ▶ A plateia só pode participar se o piloto 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.



Coding dojo - Randori Kata

Funcionamento:

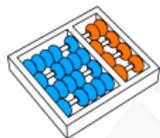
- ▶ O mestre propõe um problema.
- ▶ O piloto e o copiloto tentam solucioná-lo durante 5 minutos:
 - ▶ Piloto e copiloto devem explicar a ideia de solução para a plateia.
 - ▶ Só o piloto pode programar.
 - ▶ O copiloto pode apontar erros e dar sugestões.
 - ▶ A plateia só pode participar se o piloto 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.



Coding dojo - Randori Kata

Funcionamento:

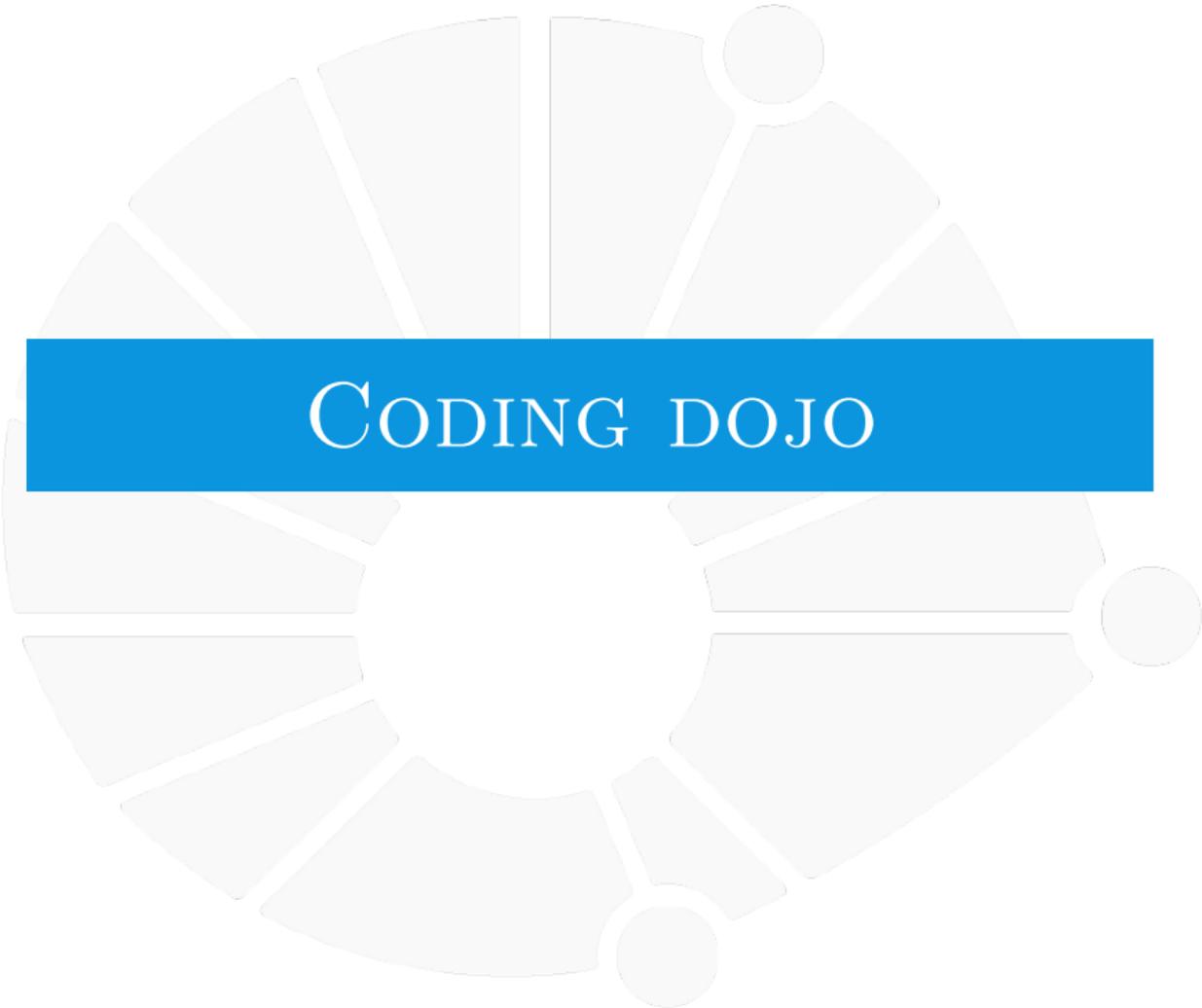
- ▶ O mestre propõe um problema.
- ▶ O piloto e o copiloto tentam solucioná-lo durante 5 minutos:
 - ▶ Piloto e copiloto devem explicar a ideia de solução para a plateia.
 - ▶ Só o piloto pode programar.
 - ▶ O copiloto pode apontar erros e dar sugestões.
 - ▶ A plateia só pode participar se o piloto 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.



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.



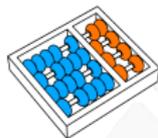
CODING DOJO



Prontos para começar?

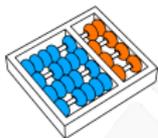


**CODE
DOJO**



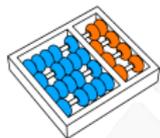
Elemento mais repetido

Faça uma função que dada uma matriz de elementos, em que não há repetições em uma mesma linha, retorne o elemento que mais aparece na matriz.



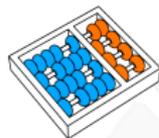
Todos os pares que somam

Faça uma função que dada uma lista de inteiro e um número, retorna alguma estrutura que armazena todos os pares da lista que somados são iguais ao número dado.



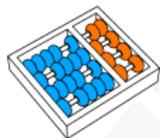
Anagramas

- ▶ Os anagramas de uma palavra são outras palavras que se obtém permutando a ordem das letras. Por exemplo “ALMA”, “LAMA” e “MALA” são anagramas.
- ▶ Faça uma função que dadas duas strings determine se uma é anagrama de outra.
- ▶ Usando a função anterior, faça uma segunda função que recebe um conjunto de strings e retorna uma estrutura em que as strings estejam separadas por grupos de anagramas.



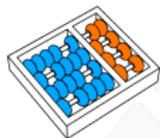
Quadrado máximo

Faça uma função que dada uma matriz binária, retorna o tamanho do maior quadrado de 1's.



Maximizando a soma

- ▶ Faça uma função que recebe uma matriz de inteiros e duas posições adjacentes na matriz (numa mesma linha ou coluna). A sua função deve multiplicar o valor de cada uma das posições por -1 e substituir, nas respectivas posições da matriz da entrada.
- ▶ Faça uma função que recebe uma matriz de inteiros e retorna a soma dos elementos.
- ▶ Faça uma função que recebe uma matriz de inteiros e aplica a primeira função tantas vezes quanto necessário, de forma a maximizar o valor da segunda função.



Mínimo número de alterações

Faça uma função que, dadas duas palavras, retorna o menor número de inserções, remoções ou trocas de letras que a segunda precisa fazer para ficar igual à primeira.

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)

05/25

17



UNICAMP

