Bubblesort em Pseudocodigo

Fonte: chat gpt

1. Declare um array com 8 números a serem ordenados

2. Inicie um loop externo de 0 até o tamanho do array - 1

3. Inicie um loop interno de 0 até o tamanho do array - 1 - i

4. Se o elemento atual for maior que o próximo elemento, troque-os de posição

5. Repita os passos 3-4 até que o loop interno esteja completo

6. Repita os passos 2-5 até que o loop externo esteja completo

7. Exiba o array ordenado

A prof. Lucy informou um erro de sintaxe no codigo do chat gpt:

// Declaração do array com 8 números a serem ordenados

array = [7, 3, 1, 5, 2, 8, 4, 6]

// Loop externo para percorrer todo o array

for i de 0 até o tamanho do array - 1

// Loop interno para comparar cada elemento com o próximo

for j de 0 até o tamanho do array - 1 - i

// Se o elemento atual for maior que o próximo, troca as posições

se array[j] > array[j+1]

temp = array[j]

array[j] = array[j+1]

array[j+1] = temp

fim se

fim loop interno

fim loop externo

// Exibe o array ordenado

exibir array

Codigo corrigido:

// Declaração do array com 8 números a serem ordenados

array = [7, 3, 1, 5, 2, 8, 4, 6]

// Loop externo para percorrer todo o array

for i de 0 até o tamanho do array - 2

// para testar depois

for i de 1 até o tamanho do array - 1

// Loop interno para comparar cada elemento com o próximo

for j de 0 até o tamanho do array - 1 - i

// Se o elemento atual for maior que o próximo, troca as posições

se array[j] > array[j+1]

temp = array[j] // temp é nossa variável temporária

array[j] = array[j+1]

array[j+1] = temp

fim se

fim loop interno

fim loop externo

// Exibe o array ordenado

exibir array

VETOR

[7, 3, 1, 5, 2, 8, 4, 6]

[0, 1, 2, 3, 4, 5, 6, 7] - 8 ELEMENTOS

0 ATE 8 ( QUE É O TAMANHO DO ARRAY) - 1 = 7

PARA COMPARAR 7 ELEMENTOS, USAMOS 6 COMP

for j de 0 até o tamanho do array - 1 - i (SABEMOS QUE i é 1)

| i | j | array j | array j + 1 | vetor |
| --- | --- | --- | --- | --- |
| 1 | 0 | 7 | 3 | [7, 3, 1, 5, 2, 8, 4, 6] |
| 1 | 0 | 3 | 7 | [3, 7, 1, 5, 2, 8, 4, 6] |
| 1 | 1 | 7 | 1 | [3, 7, 1, 5, 2, 8, 4, 6] |
| 1 | 1 | 1 | 7 | [3, 1, 7, 5, 2, 8, 4, 6] |
| 1 | 2 | 7 | 5 | [3, 1, 7, 5, 2, 8, 4, 6] |
| 1 | 2 | 5 | 7 | [3, 1, 5, 7, 2, 8, 4, 6] |
| 1 | 3 | 7 | 2 | [3, 1, 5, 7, 2, 8, 4, 6] |
| 1 | 3 | 2 | 7 | [3, 1, 5, 2, 7, 8, 4, 6] |
| 1 | 4 | 7 | 8 | [3, 1, 5, 2, 7, 8, 4, 6] |
| 1 | 4 | 7 | 8 | [3, 1, 5, 2, 7, 8, 4, 6] |
| 1 | 5 | 8 | 4 | [3, 1, 5, 2, 7, 8, 4, 6] |
| 1 | 5 | 4 | 8 | [3, 1, 5, 2, 7, 4, 8, 6] |
| 1 | 6 | 8 | 6 | [3, 1, 5, 2, 7, 4, 8, 6] |
| 1 | 6 | 6 | 8 | [3, 1, 5, 2, 7, 4, 6, 8] |
| 2 | 0 | 3 | 1 | [3, 1, 5, 2, 7, 4, 6, 8] |
| 2 | 0 | 1 | 3 | [1, 3, 5, 2, 7, 4, 6, 8] |
| 2 | 1 | 3 | 5 | [1, 3, 5, 2, 7, 4, 6, 8] |
| 2 | 1 | 3 | 5 | [1, 3, 5, 2, 7, 4, 6, 8] |
| 2 | 2 | 5 | 2 | [1, 3, 5, 2, 7, 4, 6, 8] |
| 2 | 2 | 2 | 5 | [1, 3, 2, 5, 7, 4, 6, 8] |

| 2 | 3 | 5 | 7 | [1, 3, 2, 5, 7, 4, 6, 8] |
| --- | --- | --- | --- | --- |
| 2 | 3 | 5 | 7 | [1, 3, 2, 5, 7, 4, 6, 8] |
| 2 | 4 | 7 | 4 | [1, 3, 2, 5, 7, 4, 6, 8] |
| 2 | 4 | 4 | 7 | [1, 3, 2, 5, 4, 7, 6, 8] |
| 2 | 5 | 7 | 6 | [1, 3, 2, 5, 4, 7, 6, 8] |
| 2 | 5 | 6 | 7 | [1, 3, 2, 5, 4, 6, 7, 8] |
| 3 | 0 | 1 | 3 | [1, 3, 2, 5, 4, 6, 7, 8] |
| 3 | 0 | 1 | 3 | [1, 3, 2, 5, 4, 6, 7, 8] |
| 3 | 1 | 3 | 2 | [1, 3, 2, 5, 4, 6, 7, 8] |
| 3 | 1 | 2 | 3 | [1, 3, 2, 5, 4, 6, 7, 8] |
| 3 | 2 | 3 | 5 | [1, 2, 3, 5, 4, 6, 7, 8] |
| 3 | 2 | 3 | 5 | [1, 2, 3, 5, 4, 6, 7, 8] |
| 3 | 3 | 5 | 4 | [1, 2, 3, 5, 4, 6, 7, 8] |
| 3 | 3 | 4 | 5 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 3 | 4 | 5 | 6 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 3 | 4 | 5 | 6 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 4 | 0 | 1 | 2 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 4 | 0 | 1 | 2 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 4 | 1 | 2 | 3 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 4 | 1 | 2 | 3 | [1, 2, 3, 4, 5, 6, 7, 8] |

| 4 | 2 | 3 | 4 | [1, 2, 3, 4, 5, 6, 7, 8] |
| --- | --- | --- | --- | --- |
| 4 | 2 | 3 | 4 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 4 | 3 | 4 | 2 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 4 | 3 | 4 | 5 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 5 | 0 | 1 | 2 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 5 | 0 | 1 | 2 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 5 | 1 | 2 | 3 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 5 | 1 | 2 | 3 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 5 | 2 | 3 | 4 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 5 | 2 | 3 | 4 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 6 | 0 | 1 | 2 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 6 | 0 | 1 | 2 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 6 | 1 | 2 | 3 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 6 | 1 | 2 | 3 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 7 | 0 | 1 | 2 | [1, 2, 3, 4, 5, 6, 7, 8] |
| 7 | 0 | 1 | 2 | [1, 2, 3, 4, 5, 6, 7, 8] |

Tarefa:

Até 17 de março

Quantos números tem o seu RM?

* 8 Números

Fazer um bubblesort do seu RM com um teste de mesa (seguindo o exemplo acima)