

Exercícios de Revisão

Aluno: Gabriela Stéfane dos Reis

RA:046051

Missão 4

Considere os seguintes cenários e responda qual a complexidade do algoritmo:

a) Programa que calcula o IMC, ou seja, usuário entra com peso e altura e o sistema retorna qual o seu IMC.

Complexidade: $O(1)$

b) Programa que percorre um vetor de notas de alunos da turma de algoritmo de tamanho N (pode variar de acordo com o tamanho da turma) e mostra a média geral da turma.

Complexidade: $O(n)$

c) Programa que percorre uma Lista de nomes de tamanho N e mostra o maior nome dessa lista.

Complexidade: $O(n)$

d) Programa que percorre uma matriz bidimensional $N \times N$ e mostra qual o maior número dessa matriz.

Complexidade: $O(n^2)$

Missão 5

a)

O algoritmo vai trabalhar agora com a partição [9, 5, 1, 30, 17, 11, 8, 23, 2, 13].

- Primeiro o algoritmo seleciona o número 9 (index 0) como pivô e o storeIndex = 1 (que no caso é a posição do index do pivô + 1).
- Verificou se o número 5 (index = 2) é menor ou igual que o pivô 9 (index 0).
- Como o 5 é menor que o pivô 9, vão trocar o índice 1 pelo valor do storeIndex que no momento era 1 e o storeIndex passa a ser o 2 agora.
- Verificou se o número 1 (index = 2) é menor ou igual que o pivô 9 (index 0).
- Como o 1 é menor que o pivô 9, vão trocar o índice 2 pelo valor do storeIndex que no momento era 2 e o storeIndex passa a ser o 3 agora.
- Verificou se o número 30 (index = 3) é menor ou igual que o pivô 9 (index 0).
- Como 30 é maior que o pivô 9 o algoritmo vai para a próxima comparação.
- Verificou se o número 17 (index = 4) é menor ou igual que o pivô 9 (index 0).
- Como 17 é maior que o pivô 9 o algoritmo vai para a próxima comparação.
- Verificou se o número 11 (index = 5) é menor ou igual que o pivô 9 (index 0).
- Como 11 é maior que o pivô 9 o algoritmo vai para a próxima comparação.
- Verificou se o número 8 (index = 6) é menor ou igual que o pivô 9 (index 0).

- Como o 8 é menor que o pivô 9, vão trocar o índice 6 pelo valor do storeIndex que no momento era 3 e o storeIndex passa a ser o 4 agora.
- Verificou se o número 23 (index = 7) é menor ou igual que o pivô 9 (index 0).
- Como 23 é maior que o pivô 9 o algoritmo vai para a próxima comparação.
- Verificou se o número 2 (index = 8) é menor ou igual que o pivô 9 (index 0).
- Como o 2 é menor que o pivô 9, vão trocar o índice 8 pelo valor do storeIndex que no momento era 4 e o storeIndex passa a ser o 5 agora.
- Verificou se o número 13 (index = 8) é menor ou igual que o pivô 9 (index 0).
- Como 13 é maior que o pivô 9 o algoritmo vai para a próxima comparação.
- Finaliza as comparações com o pivô 9.
- Troca a posição do pivô 9 (index 0) com o número 2 (index 4), essa troca de posição é feita a partir do valor do storeIndex-1 (era 5 no momento e com o - 1, resulta no index 4 que era posição do número 2).
- O pivô 9 está agora em sua posição classificada.

O algoritmo vai trabalhar agora com a partição [2, 5, 1, 8].

- Primeiro o algoritmo seleciona o número 2 (index 0) como pivô e o storeIndex = 1 (que no caso é a posição do index do pivô + 1).
- Verificou se o número 5 (index = 1) é menor ou igual que o pivô 2 (index 0).
- Como 5 é maior que o pivô 2 o algoritmo vai para a próxima comparação.
- Verificou se o número 1 (index = 2) é menor ou igual que o pivô 2 (index 0).
- Como o 1 é menor que o pivô 2, vão trocar o índice 2 pelo valor do storeIndex que no momento era 1 e o storeIndex passa a ser o 2 agora.
- Verificou se o número 8 (index = 3) é menor ou igual que o pivô 2 (index 0).
- Como 8 é maior que o pivô 2 o algoritmo vai para a próxima etapa.
- Finaliza as comparações com o pivô 2.
- Troca a posição do pivô 2 (index 0) com o número 1 (index 1), essa troca de posição é feita a partir do valor do storeIndex-1 (era 2 no momento e com o - 1, resulta no index 1 que era posição do número 1).
- O pivô 2 está agora em sua posição classificada.

O algoritmo vai trabalhar agora com a partição que os números são menores que 2.

- Como o tamanho dessa partição é igual a 1 elemento, esse elemento está classificado na sua posição. Ou seja, o número 1 está classificado no index 0.

O algoritmo vai trabalhar agora com a partição [5, 8].

- Primeiro o algoritmo seleciona o número 5 (index 2) como pivô e o storeIndex = 3 (que no caso é a posição do index do pivô + 1).
- Verificou se o número 8 (index = 3) é menor ou igual que o pivô 5 (index 2).
- Como 8 é maior que o pivô 5 o algoritmo vai para a próxima etapa.
- Finaliza as comparações com o pivô 5.
- Não há troca de posição do pivô 5 por ele já estar no lugar de classificação dele. Essa troca de posição é feita a partir do valor do storeIndex-1 (era 3 no momento e com o - 1, resulta no index 2 que já era posição do número 5).
- O pivô 5 está em sua posição classificada.

O algoritmo vai trabalhar agora com a partição [8].

- Como o tamanho dessa partição é igual a 1 elemento, esse elemento está classificado na sua posição. Ou seja, o número 8 está classificado no index 3.

O algoritmo vai trabalhar agora com a partição [11, 30, 23, 17, 13].

- Primeiro o algoritmo seleciona o número 11 (index 5) como pivô e o storeIndex = 6 (que no caso é a posição do index do pivô + 1).
- Verificou se o número 30 (index = 6) é menor ou igual que o pivô 11 (index 5).
- Como 30 é maior que o pivô 11 o algoritmo vai para a próxima etapa.
- Verificou se o número 23 (index = 7) é menor ou igual que o pivô 11 (index 5).
- Como 23 é maior que o pivô 11 o algoritmo vai para a próxima etapa.
- Verificou se o número 17 (index = 8) é menor ou igual que o pivô 11 (index 5).
- Como 17 é maior que o pivô 11 o algoritmo vai para a próxima etapa.
- Verificou se o número 13 (index = 9) é menor ou igual que o pivô 11 (index 5).
- Como 13 é maior que o pivô 11 o algoritmo vai para a próxima etapa.
- Finaliza as comparações com o pivô 5.
- Não há troca de posição do pivô 11 por ele já estar no lugar de classificação dele. Essa troca de posição é feita a partir do valor do storeIndex-1 (era 6 no momento e com o - 1, resulta no index 5 que já era posição do número 11).
- O pivô 11 está em sua posição classificada.

O algoritmo vai trabalhar agora com a partição [30, 23, 17, 13].

- Primeiro o algoritmo seleciona o número 30 (index 6) como pivô e o storeIndex = 7 (que no caso é a posição do index do pivô + 1).
- Verificou se o número 23 (index = 7) é menor ou igual que o pivô 30 (index 6).
- Como o 23 é menor que o pivô 30, vão trocar o índice 7 (número 23) pelo valor do storeIndex que no momento era 7 e o storeIndex passa a ser 8 agora.
- Verificou se o número 17 (index = 8) é menor ou igual que o pivô 30 (index 6).
- Como o 17 é menor que o pivô 30, vão trocar o índice 8 (número 17) pelo valor do storeIndex que no momento era 8 e o storeIndex passa a ser 9 agora.
- Verificou se o número 13 (index = 9) é menor ou igual que o pivô 30 (index 6).
- Como o 13 é menor que o pivô 30, vão trocar o índice 9 (número 13) pelo valor do storeIndex que no momento era 8 e o storeIndex passa a ser 10 agora.
- Finaliza as comparações com o pivô 30.
- Troca a posição do pivô 30 (index 6) com o número 13 (index 9), essa troca de posição é feita a partir do valor do storeIndex-1 (era 10 no momento e com o - 1, resulta no index 9 que era posição do número 13).
- O pivô 30 está em sua posição classificada.

O algoritmo vai trabalhar agora com a partição [13, 23, 17].

- Primeiro o algoritmo seleciona o número 13 (index 6) como pivô e o storeIndex = 7 (que no caso é a posição do index do pivô + 1).
- Verificou se o número 23 (index = 7) é menor ou igual que o pivô 13 (index 6).
- Como 23 é maior que o pivô 13 o algoritmo vai para a próxima etapa.
- Verificou se o número 17 (index = 8) é menor ou igual que o pivô 13 (index 6).
- Como 17 é maior que o pivô 13 o algoritmo vai para a próxima etapa.
- Finaliza as comparações com o pivô 13.
- Não há troca de posição do pivô 13 por ele já estar no lugar de classificação dele. Essa troca de posição é feita a partir do valor do storeIndex-1 (era 7 no momento e com o - 1, resulta no index 6 que já era posição do número 13).
- O pivô 13 está em sua posição classificada.

O algoritmo vai trabalhar agora com a partição [23, 17].

- Primeiro o algoritmo seleciona o número 23 (index 7) como pivô e o storeIndex = 8 (que no caso é a posição do index do pivô + 1).
- Verificou se o número 17 (index = 8) é menor ou igual que o pivô 23 (index 6).
- Como o 17 é menor que o pivô 23, vão trocar o índice 8 (número 17) pelo valor do storeIndex que no momento era 8 e o storeIndex passa a ser 9 agora.
- Finaliza as comparações com o pivô 23.
- Troca a posição do pivô 23 (index 7) com o número 17 (index 8), essa troca de posição é feita a partir do valor do storeIndex-1 (era 9 no momento e com o - 1, resulta no index 1 que era posição do número 17).
- O pivô 23 está em sua posição classificada.

O algoritmo vai trabalhar agora com a partição [17].

- Como o tamanho dessa partição é igual a 1 elemento, esse elemento está classificado na sua posição. Ou seja, o número 17 está classificado no index 7.

A lista está classificada!

b)

- A complexidade do Quick Sort no pior caso é $O(n^2)$. Ocorre quando o pivô divide o vetor em dois subvetores, um com zero elemento e outro com $n - 1$ elementos.
- A complexidade do Insertion Sort no pior caso é $O(n^2)$. O pior caso é a entrada está ordenada em ordem decrescente.
- A complexidade do Bubble Sort no pior caso é $O(n^2)$. O pior caso é a entrada está ordenada em ordem decrescente.