

A análise de complexidade de algoritmos considera apenas o tempo de execução como métrica de desempenho.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



A análise de complexidade de algoritmos é aplicada apenas a algoritmos específicos, não a classes de algoritmos.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



A análise de complexidade de algoritmos leva em conta apenas a capacidade computacional dos computadores em que os algoritmos serão implementados.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



A análise de complexidade não é relevante para a classificação de algoritmos quanto à sua eficiência em termos de espaço de memória ocupado.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



A análise de complexidade de algoritmos não leva em conta as restrições de armazenamento dos computadores em que serão executados.

☐ Verdadeiro

© Falso



Você conseguiu 1 de 1 pontos possíveis



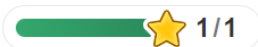
A análise de complexidade de algoritmos considera aspectos como tempo de execução, espaço de memória ocupado e energia consumida.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



A capacidade computacional e de armazenamento dos computadores onde os algoritmos serão implementados não influencia na análise de complexidade.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



A análise de complexidade leva em conta apenas o pior caso de execução de um algoritmo.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



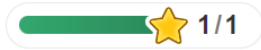
A análise de complexidade não considera a eficiência em termos de tempo de execução dos algoritmos.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



A análise de complexidade pode ser aplicada apenas a um algoritmo específico, não a uma classe de algoritmos.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



A função de complexidade representa o comportamento assintótico de um algoritmo em relação ao tamanho da entrada.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



A ordem de crescimento das funções $f_1(n) = n^2$ e $f_2(n) = n$ é $f_1(n) > f_2(n)$.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



A função $f1(n) = 2^{\{n\}}$ cresce mais rápido que a função $f4(n) = (3/2)^{\{n\}}$.

☒ Verdadeiro

☐ Falso

Você conseguiu 1 de 1 pontos possíveis



A função $f5(n) = n^{\{3\}}$ cresce mais rápido que a função $f4(n) = (3/2)^{\{n\}}$.

☐ Verdadeiro

☒ Falso

Você conseguiu 1 de 1 pontos possíveis



A função $f6(n) = 1$ é a função de crescimento mais rápido entre as listadas.

☒ Verdadeiro

☐ Falso

Você conseguiu 1 de 1 pontos possíveis



A função $f9(n) = 4n$ cresce mais rápido que a função $f8(n) = 8^{\{2n\}}$.

☒ Verdadeiro

☐ Falso

Você conseguiu 1 de 1 pontos possíveis



A classe de algoritmos exponenciais é caracterizada por um crescimento polinomial em relação ao tamanho da entrada.

☐ Verdadeiro

☒ Falso

Você conseguiu 1 de 1 pontos possíveis



Algoritmos de complexidade constante têm um crescimento mais rápido do que algoritmos de complexidade logarítmica.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



A classe de algoritmos lineares possui um crescimento linear em relação ao tamanho da entrada.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



Algoritmos exponenciais são considerados ineficientes em termos de tempo de execução.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



A classe de algoritmos logarítmicos possui um crescimento logarítmico em relação ao tamanho da entrada.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



A classe de algoritmos quadráticos possui um crescimento quadrático em relação ao tamanho da entrada.

☒ Verdadeiro

☐ Falso

Você conseguiu 1 de 1 pontos possíveis



Algoritmos de complexidade fatorial são mais eficientes em termos de espaço de memória ocupado.

☐ Verdadeiro

☒ Falso

Você conseguiu 1 de 1 pontos possíveis



Algoritmos de complexidade exponencial têm um crescimento mais rápido do que algoritmos de complexidade linear.

☐ Verdadeiro

☒ Falso

Você conseguiu 1 de 1 pontos possíveis



Algoritmos de complexidade quadrática têm um crescimento mais rápido do que algoritmos de complexidade linear.

☐ Verdadeiro

☒ Falso

Você conseguiu 1 de 1 pontos possíveis



A notação O é usada para estabelecer um limite superior para o crescimento assintótico de uma função ou algoritmo.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



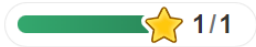
A função $3n^2 + 5n + 1$ é $\Omega(n^3)$.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



Para provar que $3n^2 + 5n + 1$ é $\Theta(n^2)$, devemos encontrar valores de c_1 , c_2 e m tais que $c_1 \times |f(n)| \leq |g(n)| \leq c_2 \times |f(n)|$ para $n \geq m$.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



A função $3n^2 + 5n + 1$ é $\Theta(n)$.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



A função $3n^2 + 5n + 1$ é $\Theta(n^3)$.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



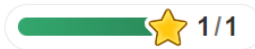
A definição das notações Θ , O e Ω envolve o uso de constantes.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



A notação Ω é usada para estabelecer um limite inferior para o crescimento assintótico de uma função ou algoritmo.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



As notações Θ , O e Ω são utilizadas para estabelecer limites superior, inferior e justo para o crescimento de funções e algoritmos.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



A notação Ω é utilizada para estabelecer um limite superior para o crescimento de funções e algoritmos.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



A notação O é utilizada para estabelecer um limite inferior para o crescimento de funções e algoritmos.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



A função de complexidade de tempo para encontrar o maior e o menor valores em um array de inteiros é $O(n)$.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



Para provar que $3n^2 + 5n + 1$ é $\Omega(n^2)$, devemos encontrar valores de c e m tais que $|g(n)| \geq c \times |f(n)|$ para $n \geq m$.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



A função $3n^2 + 5n + 1$ é $\Omega(n)$.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



No caso em que desejamos efetuar menos de n pesquisas em um array de números, a solução mais eficiente é executar uma pesquisa sequencial para cada valor desejado.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



XLIII. No caso em que desejamos efetuar n pesquisas em um array de números reais, a solução mais eficiente é ordenar o array e aplicar uma pesquisa binária.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



O algoritmo de ordenação por inserção possui complexidade $O(n^2)$ no pior caso.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



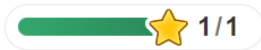
A pesquisa binária em um vetor ordenado possui complexidade $O(\log n)$.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



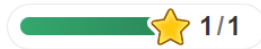
O algoritmo de ordenação por seleção é sempre mais eficiente que o algoritmo de ordenação por inserção.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



A ordenação por fusão (merge sort) tem complexidade $O(n \log n)$ no pior caso.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



A pesquisa linear em uma lista desordenada possui complexidade $O(n)$.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



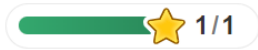
O algoritmo de ordenação quicksort possui complexidade $O(n^2)$ no pior caso.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis



A pesquisa em árvore binária de busca tem complexidade $O(n)$ no pior caso.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



Algoritmos polinomiais pertencem à classe P e são considerados algoritmos eficientes em termos de tempo de execução.

☒ Verdadeiro



☐ Falso

Você conseguiu 1 de 1 pontos possíveis



A questão P versus NP aborda a solucionabilidade eficiente de problemas computacionais, considerando-se apenas a eficiência em termos de tempo de execução.

☐ Verdadeiro

☒ Falso



Você conseguiu 1 de 1 pontos possíveis

