Aula 06

Engenharia da Computação – 3ª série

Notações Assintóticas

(E1, E2)

2024

Notações Assintóticas

Pergunta

- O que é Ordem de Grandeza de Execução?

Ordem de Grandeza de Execução

Resposta

- Por exemplo, o tempo exato de execução de um algoritmo pode ser dado pela função polinomial **$f(n) = 3n^2 + 2n + 3$**;

- Neste caso, o tempo aproximado de execução será uma função de **n^2**, ou seja **$f(n^2)$**, a mais alta potência de **n**;

- Dessa forma, pode-se desprezar o coeficiente de **n^2**, bem como os outros termos da função polinomial que define a complexidade do algoritmo;

Ordem de Grandeza de Execução

Resposta

- Assim, para efeito de análise de algoritmos, utiliza-se uma notação que seja capaz de exprimir a ordem de grandeza do tempo de execução;

- Essa notação é assintótica, ou seja, representa uma linha que se aproxima da função de complexidade do algoritmo.

Notações Assintóticas

Pergunta

- O que é Notação Big-Oh?

Notação Big-Oh

Resposta

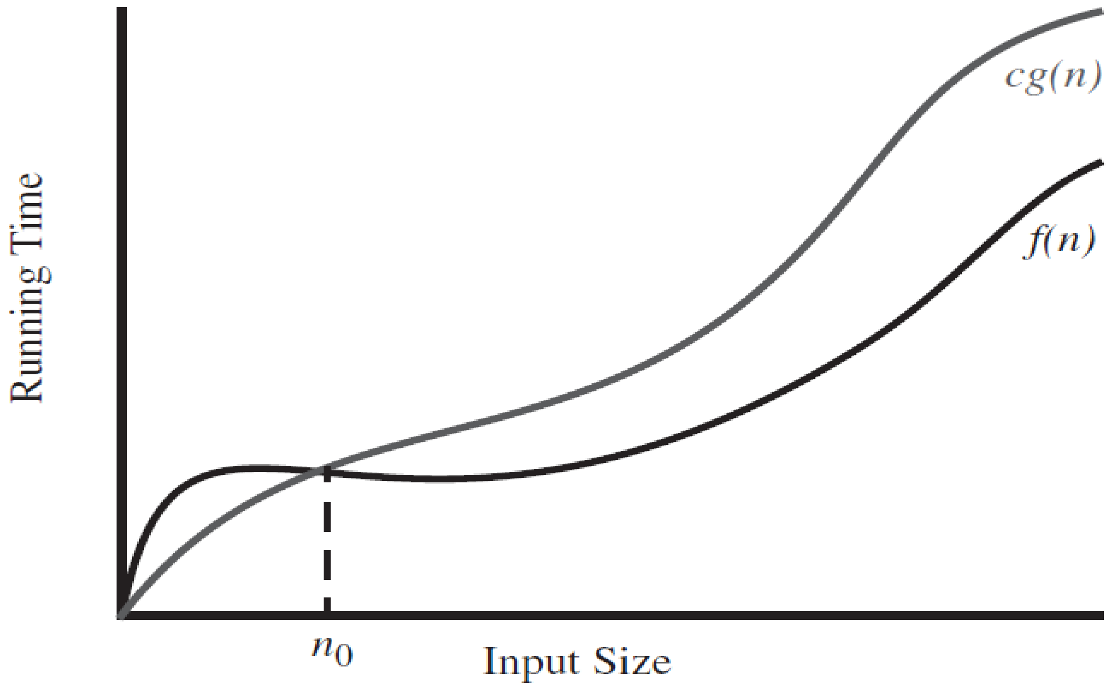
- Seja **f(n)** e **g(n)** funções que mapeiam inteiros não negativos para números reais;

- Diz-se que **f(n)** é **O(g(n))** se existir uma constante real **$c > 0$** e uma constante inteira **$n_o \geq 1$** tal que **$f(n) \leq c\, g(n)$**, para todo inteiro **$n \geq n_0$** ;

- Essa definição é frequentemente dita "**f(n)** é **Big-Oh** de **g(n)**" ou "**f(n)** é **ordem g(n)**".

Notação Big-Oh

Resposta



The function $f(n)$ is $O(g(n))$, for $f(n) \leq c \cdot g(n)$ when $n \geq n_0$.

Notações Assintóticas

Pergunta

- O que é Ordem de Complexidade Big-Oh?

Ordem de Complexidade Big-Oh

Resposta

- A notação **Big-Oh** permite que se diga que uma função de **n** é “menor ou igual” a outra função, por um fator constante, ou seja, **c** na definição;

- A notação **Big-Oh** é largamente empregada para caracterizar limites de tempo e de espaço do algoritmo em termos de um parâmetro **n**, o qual representa o tamanho do problema;

- A notação **Big-Oh** fornece limites superiores de funções que, por sua vez, correspondem ao tempo de execução de algoritmos.

Notações Assintóticas

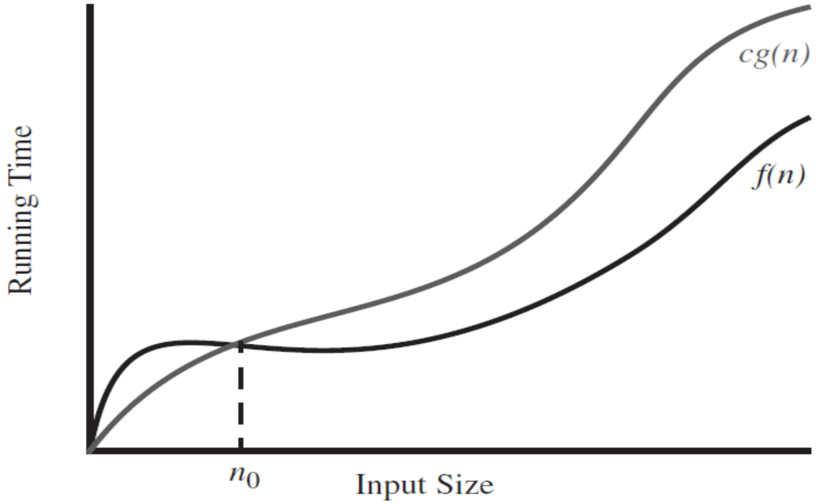
Pergunta

- A função de complexidade **F(n) = 3n + 8** é **O(n)** ?

F(n) = 3n + 8 é O(n) ?

Resposta

- Diz-se que **f(n)** é **O(g(n))** se existir uma constante real **c > 0** e uma constante inteira $\mathbf{n_o} \geq$ **1** tal que **f(n)** $\leq$ **c g(n)** para todo inteiro $\mathbf{n \geq n_0}$;



The function $f(n)$ is $O(g(n))$, for $f(n) \leq c \cdot g(n)$ when $n \geq n_0$.

F(n) = 3n + 8 é O(n) ?

Resposta

- Necessita-se de uma constante real **$c > 0$** e uma constante inteira **$n_0 \geq 1$** tal que **$3n + 8 \leq c\,n$** para todo inteiro **$n \geq n_0$** ;

$F(n) = 3n + 8$ é $O(n)$?

Resposta

- Necessita-se de uma constante real $\mathbf{c > 0}$ e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{3n + 8 \leq c\,n}$ para todo inteiro $\mathbf{n \geq n_0}$;
- Majorando-se a função **F(n)**, tem-se:

$$F(n) = 3n + 8 \leq 3n + 8n$$

F(n) = 3n + 8 é O(n) ?

Resposta

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_0} \geq$ **1** tal que **3n + 8 ≤ c n** para todo inteiro $\mathbf{n} \geq \mathbf{n_0}$;
- Majorando-se a função **F(n)**, tem-se:

$$F(n) = 3n + 8 \leq 3n + 8n$$

- Portanto, **F(n) = 3n + 8 ≤ 3n + 8n ≤ 11n → c n** (c: constante)
- A expressão acima é verdadeira para todo **n > 0**;

F(n) = 3n + 8 é O(n) ?

Resposta

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_0} \geq$ **1** tal que **3n + 8 ≤ c n** para todo inteiro $\mathbf{n} \geq \mathbf{n_0}$;

- Majorando-se a função **F(n)**, tem-se:

$$F(n) = 3n + 8 \leq 3n + 8n$$

- Portanto, **F(n) = 3n + 8 ≤ 3n + 8n ≤ 11n → c n** (c: constante)

- A expressão acima é verdadeira para todo **n > 0**;

- Logo, existe **c = 11**, tal que **F(n) = 3n + 8 ≤ c n**, para todo **n > 0**;

F(n) = 3n + 8 é O(n) ?

Resposta

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{3n + 8 \leq c\,n}$ para todo inteiro $\mathbf{n \geq n_0}$;

- Majorando-se a função **F(n)**, tem-se:

$$\mathbf{F(n) = 3n + 8 \leq 3n + 8n}$$

- Portanto, $\mathbf{F(n) = 3n + 8 \leq 3n + 8n \leq 11n \rightarrow c\,n}$ (c: constante)

- A expressão acima é verdadeira para todo **n > 0**;

- Logo, existe **c = 11**, tal que $\mathbf{F(n) = 3n + 8 \leq c\,n}$, para todo **n > 0**;

- Assim, **3n + 8** é **O(n)**.

Notações Assintóticas

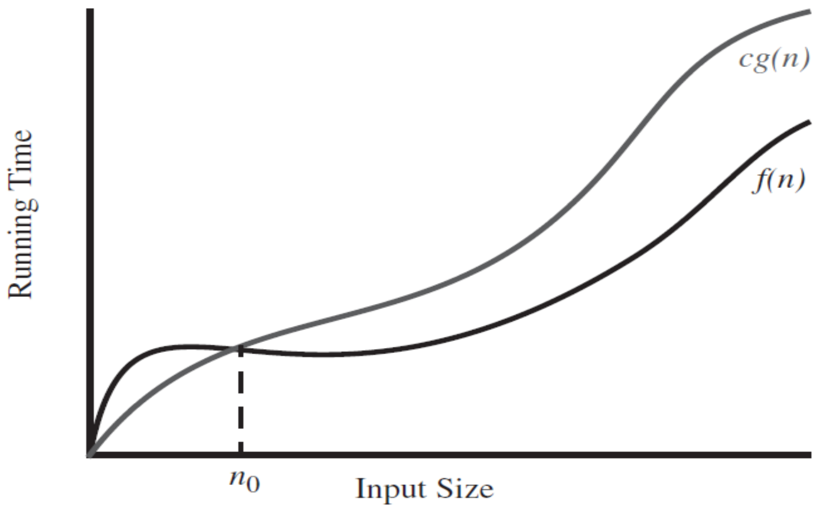
Pergunta

- A função de complexidade **$F(n) = 3n + 8$** é **$O(n^2)$** ?

$F(n) = 3n + 8$ é $O(n^2)$?

Resposta

- Diz-se que **f(n)** é **O(g(n))** se existir uma constante real **c > 0** e uma constante inteira $\mathbf{n_o \geq 1}$ tal que $\mathbf{f(n) \leq c\, g(n)}$ para todo inteiro $\mathbf{n \geq n_0}$;



The function $f(n)$ is $O(g(n))$, for $f(n) \leq c \cdot g(n)$ when $n \geq n_0$.

$F(n) = 3n + 8$ é $O(n^2)$?

<u>Resposta</u>

- Necessita-se de uma constante real **$c > 0$** e uma constante inteira **$n_0 \geq 1$** tal que **$3n + 8 \leq c\, n^2$** para todo inteiro **$n \geq n_0$** ;

F(n) = 3n + 8 é O(n²) ?

<u>Resposta</u>

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{3n + 8 \leq c\,n^2}$ para todo inteiro $\mathbf{n \geq n_0}$;

- Majorando-se a função **F(n)**, tem-se:

$$\mathbf{F(n) = 3n + 8 \leq 3n^2 + 8n^2}$$

$F(n) = 3n + 8$ é $O(n^2)$?

<u>Resposta</u>

- Necessita-se de uma constante real **$c > 0$** e uma constante inteira **$n_0 \geq 1$** tal que **$3n + 8 \leq c\,n^2$** para todo inteiro **$n \geq n_0$** ;

- Majorando-se a função **F(n)**, tem-se:

$$F(n) = 3n + 8 \leq 3n^2 + 8n^2$$

- Portanto, **$F(n) = 3n + 8 \leq 3n^2 + 8n^2 \leq 11n^2 \rightarrow c\,n^2$** (c: constante)

- A expressão acima é verdadeira para todo **$n > 0$**;

$F(n) = 3n + 8$ é $O(n^2)$?

Resposta

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{3n + 8 \leq c\,n^2}$ para todo inteiro $\mathbf{n \geq n_0}$;
- Majorando-se a função **F(n)**, tem-se:

$$\mathbf{F(n) = 3n + 8 \leq 3n^2 + 8n^2}$$

- Portanto, $\mathbf{F(n) = 3n + 8 \leq 3n^2 + 8n^2 \leq 11n^2 \rightarrow c\,n^2}$ (c: constante)
- A expressão acima é verdadeira para todo **n > 0**;
- Logo, existe **c = 11**, tal que $\mathbf{F(n) = 3n + 8 \leq c\,n^2}$, para todo **n > 0**;

ECM306 – Tópicos Avançados em Estrutura de Dados

F(n) = 3n + 8 é O(n²) ?

Resposta

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{3n + 8 \leq c\, n^2}$ para todo inteiro $\mathbf{n \geq n_0}$;
- Majorando-se a função **F(n)**, tem-se:

$$\mathbf{F(n) = 3n + 8 \leq 3n^2 + 8n^2}$$

- Portanto, $\mathbf{F(n) = 3n + 8 \leq 3n^2 + 8n^2 \leq 11n^2 \rightarrow c\, n^2}$ (c: constante)
- A expressão acima é verdadeira para todo **n > 0**;
- Logo, existe **c = 11**, tal que $\mathbf{F(n) = 3n + 8 \leq c\, n^2}$, para todo **n > 0**;
- Assim, **3n + 8** é $\mathbf{O(n^2)}$.

Notações Assintóticas

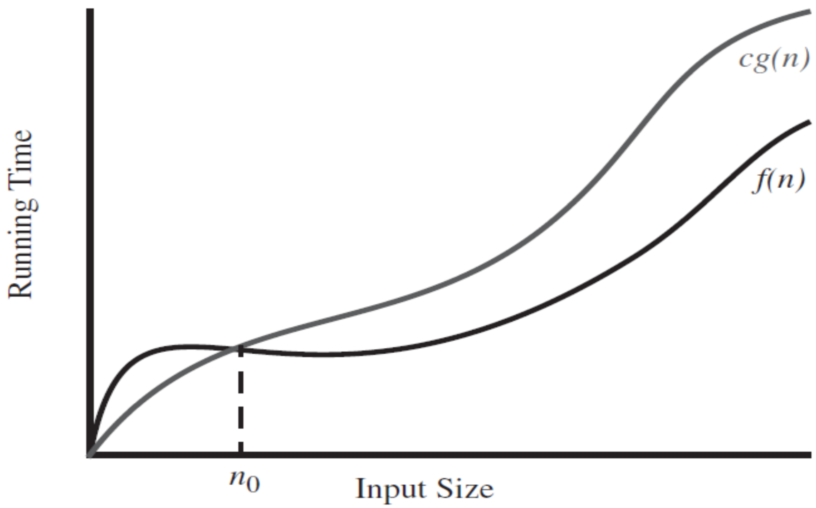
Pergunta

- A função de complexidade **F(n) = 3n + 8** é **O(n³)** ?

$F(n) = 3n + 8$ é $O(n^3)$?

Resposta

- Diz-se que **f(n)** é **O(g(n))** se existir uma constante real **c > 0** e uma constante inteira $\mathbf{n_o \geq 1}$ tal que $\mathbf{f(n) \leq c\ g(n)}$ para todo inteiro $\mathbf{n \geq n_0}$;



The function $f(n)$ is $O(g(n))$, for $f(n) \leq c \cdot g(n)$ when $n \geq n_0$.

$F(n) = 3n + 8$ é $O(n^3)$?

<u>Resposta</u>

- Necessita-se de uma constante real **$c > 0$** e uma constante inteira **$n_0 \geq 1$** tal que **$3n + 8 \leq c\,n^3$** para todo inteiro **$n \geq n_0$** ;

$F(n) = 3n + 8$ é $O(n^3)$?

Resposta

- Necessita-se de uma constante real **$c > 0$** e uma constante inteira **$n_0 \geq 1$** tal que **$3n + 8 \leq c\, n^3$** para todo inteiro **$n \geq n_0$** ;
- Majorando-se a função **F(n)**, tem-se:

$$F(n) = 3n + 8 \leq 3n^3 + 8n^3$$

$F(n) = 3n + 8$ é $O(n^3)$?

Resposta

- Necessita-se de uma constante real **$c > 0$** e uma constante inteira **$n_0 \geq 1$** tal que **$3n + 8 \leq c\,n^3$** para todo inteiro **$n \geq n_0$** ;

- Majorando-se a função **F(n)**, tem-se:

$$F(n) = 3n + 8 \leq 3n^3 + 8n^3$$

- Portanto, **$F(n) = 3n + 8 \leq 3n^3 + 8n^3 \leq 11n^3 \rightarrow c\,n^3$** (c: constante)

- A expressão acima é verdadeira para todo **$n > 0$**;

$F(n) = 3n + 8$ é $O(n^3)$?

Resposta

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{3n + 8 \leq c\, n^3}$ para todo inteiro $\mathbf{n \geq n_0}$;
- Majorando-se a função **F(n)**, tem-se:

$$F(n) = 3n + 8 \leq 3n^3 + 8n^3$$

- Portanto, $\mathbf{F(n) = 3n + 8 \leq 3n^3 + 8n^3 \leq 11n^3 \rightarrow c\, n^3}$ (c: constante)
- A expressão acima é verdadeira para todo **n > 0**;
- Logo, existe **c = 11**, tal que $\mathbf{F(n) = 3n + 8 \leq c\, n^3}$, para todo **n > 0**;

F(n) = 3n + 8 é O(n³) ?

<u>Resposta</u>

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{3n + 8 \leq c\,n^3}$ para todo inteiro $\mathbf{n \geq n_0}$;

- Majorando-se a função **F(n)**, tem-se:

$$\mathbf{F(n) = 3n + 8 \leq 3n^3 + 8n^3}$$

- Portanto, $\mathbf{F(n) = 3n + 8 \leq 3n^3 + 8n^3 \leq 11n^3 \rightarrow c\,n^3}$ (c: constante)

- A expressão acima é verdadeira para todo **n > 0**;

- Logo, existe **c = 11**, tal que $\mathbf{F(n) = 3n + 8 \leq c\,n^3}$, para todo **n > 0**;

- Assim, **3n + 8** é $\mathbf{O(n^3)}$.

Notações Assintóticas

Conclusão

- **$3n + 8$** é **$O(n)$**;
- **$3n + 8$** é **$O(n^2)$**; e
- **$3n + 8$** é **$O(n^3)$**.
- Portanto, **$3n + 8$** $\in$ a um conjunto de funções que atendem à definição de **$O(f(n))$**.

Notações Assintóticas

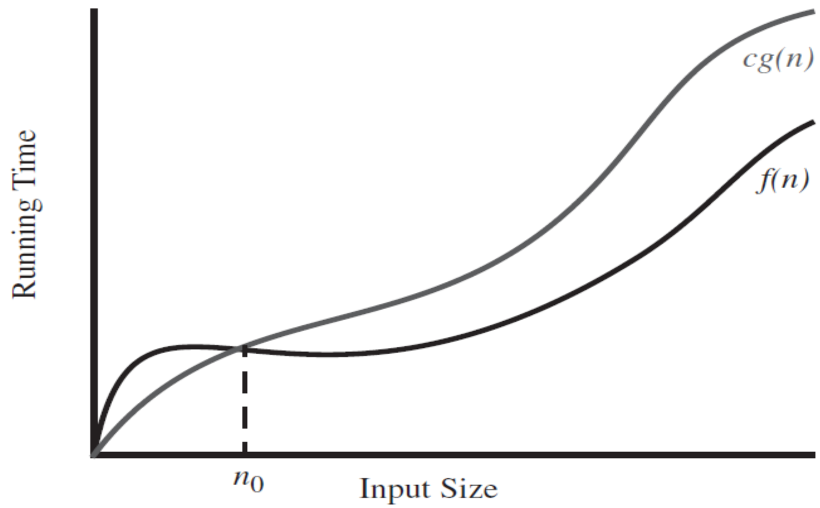
Pergunta

- A função de complexidade **$F(n) = 2n^2 + 3n + 4$** é **$O(n^2)$** ?

$F(n) = 2n^2 + 3n + 4$ é $O(n^2)$?

Resposta

- Diz-se que **f(n)** é **O(g(n))** se existir uma constante real **c > 0** e uma constante inteira **$n_o \geq 1$** tal que **f(n) ≤ c g(n)** para todo inteiro **$n \geq n_0$** ;



The function $f(n)$ is $O(g(n))$, for $f(n) \leq c \cdot g(n)$ when $n \geq n_0$.

$F(n) = 2n^2 + 3n + 4$ é $O(n^2)$?

Resposta

- Necessita-se de uma constante real $\mathbf{c > 0}$ e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{2n^2 + 3n + 4 \;\; \leq \;\; c\, n^2}$ para todo inteiro $\mathbf{n \geq n_0}$;

$F(n) = 2n^2 + 3n + 4$ é $O(n^2)$?

Resposta

- Necessita-se de uma constante real **$c > 0$** e uma constante inteira **$n_0 \geq 1$** tal que **$2n^2 + 3n + 4 \leq c\, n^2$** para todo inteiro **$n \geq n_0$** ;
- Majorando-se a função **F(n)**, tem-se:

$$F(n) = 2n^2 + 3n + 4 \leq 2n^2 + 3n^2 + 4n^2$$

$F(n) = 2n^2 + 3n + 4$ é $O(n^2)$?

Resposta

- Necessita-se de uma constante real **$c > 0$** e uma constante inteira **$n_0 \geq 1$** tal que **$2n^2 + 3n + 4 \leq c\, n^2$** para todo inteiro **$n \geq n_0$** ;

- Majorando-se a função **$F(n)$**, tem-se:

$$F(n) = 2n^2 + 3n + 4 \leq 2n^2 + 3n^2 + 4n^2$$

- Portanto, **$F(n) = 2n^2 + 3n + 4 \leq 2n^2 + 3n^2 + 4n^2 \leq 9n^2 \rightarrow c\, n^2$**

- A expressão acima é verdadeira para todo **$n > 0$**;

$F(n) = 2n^2 + 3n + 4$ é $O(n^2)$?

Resposta

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{2n^2 + 3n + 4 \leq c\, n^2}$ para todo inteiro $\mathbf{n \geq n_0}$;
- Majorando-se a função **F(n)**, tem-se:

$$\mathbf{F(n) = 2n^2 + 3n + 4 \leq 2n^2 + 3n^2 + 4n^2}$$

- Portanto, $\mathbf{F(n) = 2n^2 + 3n + 4 \leq 2n^2 + 3n^2 + 4n^2 \leq 9n^2 \rightarrow c\, n^2}$
- A expressão acima é verdadeira para todo **n > 0**;
- Logo, existe **c = 9**, tal que $\mathbf{2n^2 + 3n + 4 \leq c\, n^2}$, para todo **n > 0**;

$F(n) = 2n^2 + 3n + 4$ é $O(n^2)$?

Resposta

- Necessita-se de uma constante real $\mathbf{c > 0}$ e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{2n^2 + 3n + 4 \leq c\, n^2}$ para todo inteiro $\mathbf{n \geq n_0}$;
- Majorando-se a função **F(n)**, tem-se:

$$F(n) = 2n^2 + 3n + 4 \leq 2n^2 + 3n^2 + 4n^2$$

- Portanto, $\mathbf{F(n) = 2n^2 + 3n + 4 \leq 2n^2 + 3n^2 + 4n^2 \leq 9n^2 \rightarrow c\, n^2}$
- A expressão acima é verdadeira para todo $\mathbf{n > 0}$;
- Logo, existe $\mathbf{c = 9}$, tal que $\mathbf{2n^2 + 3n + 4 \leq c\, n^2}$, para todo $\mathbf{n > 0}$;
- Assim, $\mathbf{2n^2 + 3n + 4}$ é $\mathbf{O(n^2)}$.

Notações Assintóticas

Pergunta

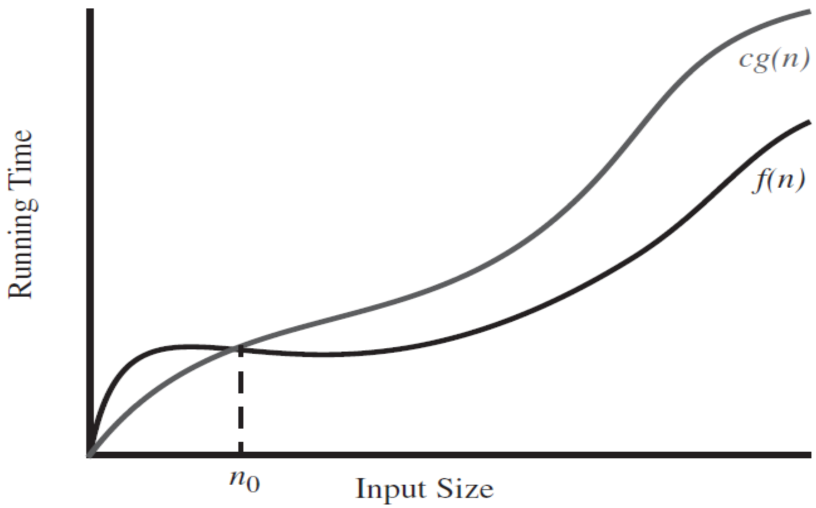
- A função de complexidade **$F(n) = 2n^2 + 3n + 4$** é **$O(n^3)$** ?

ECM306 – Tópicos Avançados em Estrutura de Dados

$F(n) = 2n^2 + 3n + 4$ é $O(n^3)$?

Resposta

- Diz-se que **f(n)** é **O(g(n))** se existir uma constante real **c > 0** e uma constante inteira **$n_o \geq 1$** tal que **f(n) ≤ c g(n)** para todo inteiro **$n \geq n_0$** ;



The function $f(n)$ is $O(g(n))$, for $f(n) \leq c \cdot g(n)$ when $n \geq n_0$.

F(n) = $2n^2 + 3n + 4$ é $O(n^3)$?

Resposta

- Necessita-se de uma constante real **$c > 0$** e uma constante inteira **$n_0 \geq 1$** tal que **$2n^2 + 3n + 4 \leq c\, n^3$** para todo inteiro **$n \geq n_0$** ;

$F(n) = 2n^2 + 3n + 4$ é $O(n^3)$?

Resposta

- Necessita-se de uma constante real **$c > 0$** e uma constante inteira **$n_0 \geq 1$** tal que **$2n^2 + 3n + 4 \leq c\,n^3$** para todo inteiro **$n \geq n_0$** ;

- Majorando-se a função **F(n)**, tem-se:

$$F(n) = 2n^2 + 3n + 4 \leq 2n^3 + 3n^3 + 4n^3$$

$F(n) = 2n^2 + 3n + 4$ é $O(n^3)$?

Resposta

- Necessita-se de uma constante real **$c > 0$** e uma constante inteira **$n_0 \geq 1$** tal que **$2n^2 + 3n + 4 \leq c\, n^3$** para todo inteiro **$n \geq n_0$** ;
- Majorando-se a função **$F(n)$**, tem-se:

$$F(n) = 2n^2 + 3n + 4 \leq 2n^3 + 3n^3 + 4n^3$$

- Portanto, **$F(n) = 2n^2 + 3n + 4 \leq 2n^3 + 3n^3 + 4n^3 \leq 9n^3 \rightarrow c\, n^3$**
- A expressão acima é verdadeira para todo **$n > 0$**;

$F(n) = 2n^2 + 3n + 4$ é $O(n^3)$?

Resposta

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{2n^2 + 3n + 4 \leq c\,n^3}$ para todo inteiro $\mathbf{n \geq n_0}$;
- Majorando-se a função **F(n)**, tem-se:

$$F(n) = 2n^2 + 3n + 4 \leq 2n^3 + 3n^3 + 4n^3$$

- Portanto, $\mathbf{F(n) = 2n^2 + 3n + 4 \leq 2n^3 + 3n^3 + 4n^3 \leq 9n^3 \rightarrow c\,n^3}$
- A expressão acima é verdadeira para todo **n > 0**;
- Logo, existe **c = 9**, tal que $\mathbf{2n^2 + 3n + 4 \leq c\,n^3}$, para todo **n > 0**;

$F(n) = 2n^2 + 3n + 4$ é $O(n^3)$?

Resposta

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{2n^2 + 3n + 4 \leq c\, n^3}$ para todo inteiro $\mathbf{n \geq n_0}$;
- Majorando-se a função **F(n)**, tem-se:

$$F(n) = 2n^2 + 3n + 4 \leq 2n^3 + 3n^3 + 4n^3$$

- Portanto, $\mathbf{F(n) = 2n^2 + 3n + 4 \leq 2n^3 + 3n^3 + 4n^3 \leq 9n^3 \rightarrow c\, n^3}$
- A expressão acima é verdadeira para todo **n > 0**;
- Logo, existe **c = 9**, tal que $\mathbf{2n^2 + 3n + 4 \leq c\, n^3}$, para todo **n > 0**;
- Assim, $\mathbf{2n^2 + 3n + 4}$ é $\mathbf{O(n^3)}$.

Notações Assintóticas

Conclusão

- $\mathbf{2n^2 + 3n + 4}$ é $\mathbf{O(n^2)}$; e
- $\mathbf{2n^2 + 3n + 4}$ é $\mathbf{O(n^3)}$.
- Portanto, $\mathbf{2n^2 + 3n + 4}$ $\in$ a um conjunto de funções que atendem à definição de $\mathbf{O(f(n))}$;
- E somente por razões de ordem prática, prefere-se e costuma-se dizer que $\mathbf{2n^2 + 3n + 4}$ é $\mathbf{O(n^2)}$.

Notações Assintóticas

Pergunta

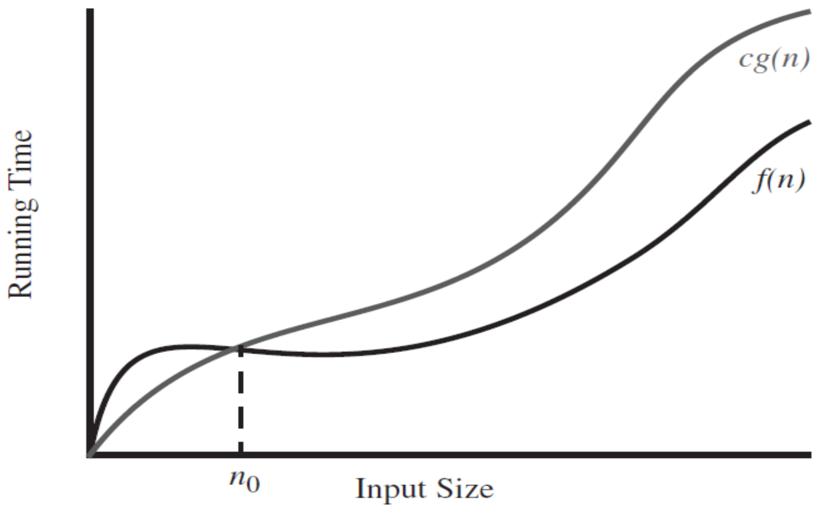
- A função de complexidade **$F(n) = 2n^2 – 3n + 4$** é **$O(n)$** ?

ECM306 – Tópicos Avançados em Estrutura de Dados

$F(n) = 2n^2 – 3n + 4$ é $O(n)$?

Resposta

- Diz-se que **f(n)** é **O(g(n))** se existir uma constante real **c > 0** e uma constante inteira **$n_o \geq 1$** tal que **$f(n) \leq c\ g(n)$** para todo inteiro **$n \geq n_0$** ;



The function $f(n)$ is $O(g(n))$, for $f(n) \leq c \cdot g(n)$ when $n \geq n_0$.

$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

Resposta

- Necessita-se de uma constante real $\mathbf{c > 0}$ e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{2n^2 - 3n + 4 \leq c\, n}$ para todo inteiro $\mathbf{n \geq n_0}$;

$F(n) = 2n^2 – 3n + 4$ é $O(n)$?

Resposta

- Necessita-se de uma constante real **$c > 0$** e uma constante inteira **$n_0 \geq 1$** tal que **$2n^2 – 3n + 4 \leq c\,n$** para todo inteiro **$n \geq n_0$** ;

- Da função **F(n)**, tem-se:

$$F(n) = 2n^2 - 3n + 4 \leq c\,n$$

$F(n) = 2n^2 – 3n + 4$ é $O(n)$?

Resposta

- Necessita-se de uma constante real $\mathbf{c > 0}$ e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{2n^2 - 3n + 4 \leq c\,n}$ para todo inteiro $\mathbf{n \geq n_0}$;
- Da função **F(n)**, tem-se:

$$F(n) = 2n^2 - 3n + 4 \leq c\,n$$

$$2n^2 - 3n + 4 + 3n \leq c\,n + 3n$$

$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

Resposta

- Necessita-se de uma constante real $\mathbf{c > 0}$ e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{2n^2 - 3n + 4 \leq c\,n}$ para todo inteiro $\mathbf{n \geq n_0}$;

- Da função **F(n)**, tem-se:

$$F(n) = 2n^2 - 3n + 4 \leq c\,n$$

$$2n^2 - 3n + 4 + 3n \leq c\,n + 3n$$

$$2n^2 - \cancel{3n} + 4 + \cancel{3n} \leq c\,n + 3n$$

$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

Resposta

- Necessita-se de uma constante real $\mathbf{c > 0}$ e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{2n^2 - 3n + 4 \leq c\,n}$ para todo inteiro $\mathbf{n \geq n_0}$;

- Da função **F(n)**, tem-se:

$$F(n) = 2n^2 - 3n + 4 \leq c\,n$$

$$2n^2 - 3n + 4 + 3n \leq c\,n + 3n$$

$$2n^2 - \cancel{3n} + 4 + \cancel{3n} \leq c\,n + 3n$$

$$2n^2 + 4 \leq c\,n + 3n$$

ECM306 – Tópicos Avançados em Estrutura de Dados

$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

Resposta

- Necessita-se de uma constante real **$c > 0$** e uma constante inteira **$n_0 \geq 1$** tal que **$2n^2 - 3n + 4 \leq c\,n$** para todo inteiro **$n \geq n_0$** ;

- Da função **F(n)**, tem-se:

$$F(n) = 2n^2 - 3n + 4 \leq c\,n$$

$$2n^2 - 3n + 4 + 3n \leq c\,n + 3n$$

$$2n^2 - \cancel{3n} + 4 + \cancel{3n} \leq c\,n + 3n$$

$$2n^2 + 4 \leq c\,n + 3n$$

$$2n^2 + 4 \leq n\,(c + 3)$$

$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

<u>Resposta</u>

- Da função **F(n)**, tem-se:

$$\mathbf{2n^2 + 4 \leq n\,(c + 3)}$$

$F(n) = 2n^2 – 3n + 4$ é $O(n)$?

Resposta

- Da função **F(n)**, tem-se:

$$\mathbf{2n^2 + 4 \;\; \leq \;\; n\,(c + 3)}$$

$$\mathbf{2n^2 + 4 \;\; \leq \;\; n\,k}$$

$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

Resposta

- Da função **F(n)**, tem-se:

$$\mathbf{2n^2 + 4 \;\leq\; n\,(c + 3)}$$

$$\mathbf{2n^2 + 4 \;\leq\; n\,k}$$

$$\mathbf{(2n^2 + 4)/n \;\leq\; (n\,k)/n}$$

ECM306 – Tópicos Avançados em Estrutura de Dados

$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

Resposta

- Da função **F(n)**, tem-se:

$$\mathbf{2n^2 + 4 \;\leq\; n\,(c + 3)}$$

$$\mathbf{2n^2 + 4 \;\leq\; n\,k}$$

$$\mathbf{(2n^2 + 4)/n \;\leq\; (n\,k)/n}$$

$$\mathbf{2n^2/n + 4/n \;\leq\; k}$$

$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

Resposta

- Da função **F(n)**, tem-se:

$$2n^2 + 4 \leq n\,(c + 3)$$

$$2n^2 + 4 \leq n\,k$$

$$(2n^2 + 4)/n \leq (n\,k)/n$$

$$2n^2/n + 4/n \leq k$$

$$2n + 4/n \leq k$$

$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

Resposta

- Da função **F(n)**, tem-se:

$$\mathbf{2n + 4/n \quad \leq \quad k}$$

$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

<u>Resposta</u>

- Da função **F(n)**, tem-se:

$$\mathbf{2n + 4/n \leq k}$$

Sendo **n** muito grande...

$$\mathbf{2n + \cancel{4/n} \leq k}$$

$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

Resposta

- Da função **F(n)**, tem-se:

$$2n + 4/n \;\; \leq \;\; k$$

Sendo **n** muito grande...

$$2n + 4\cancel{/}n \;\; \leq \;\; k$$

$$2n \;\; \leq \;\; k$$

$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

Resposta

- Da função **F(n)**, tem-se:

$$2n + 4/n \leq k$$

Sendo **n** muito grande...

$$2n + \cancel{4/n} \leq k$$

$$2n \leq k$$

$$n \leq k / 2$$

$F(n) = 2n^2 – 3n + 4$ é $O(n)$?

Resposta

- Da função **F(n)**, tem-se:

$$2n + 4/n \leq k$$

Sendo **n** muito grande...

$$2n + \cancel{4/n} \leq k$$

$$2n \leq k$$

$$n \leq k / 2$$

$$n \leq k_2$$

$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

Resposta

- Da função **F(n)**, tem-se:

$$\mathbf{n \leq k_2}$$

$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

Resposta

- Da função **F(n)**, tem-se:

$$\mathbf{n \leq k_2}$$

ABSURDO!

$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

Resposta

- Da função **F(n)**, tem-se:

$$\mathbf{n \leq k_2}$$

ABSURDO!

O valor de **n** é muito grande e, portanto, o primeiro termo da inequação nunca será inferior ao segundo termo, para qualquer $\mathbf{k_2 > 0}$ e $\mathbf{n \geq n_0}$, sendo $\mathbf{n_0 \geq 1}$.

$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

Resposta

- Da função **F(n)**, tem-se:

$$n \leq k_2$$

ABSURDO!

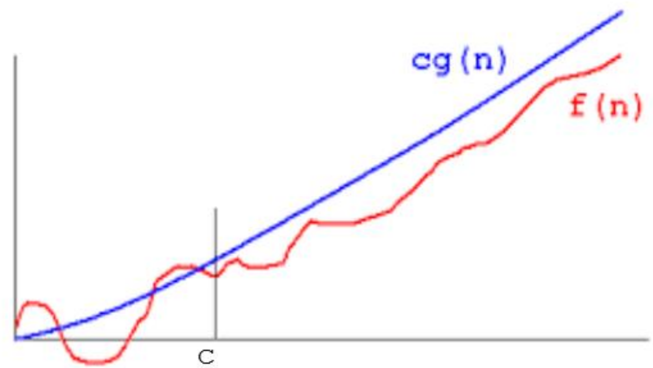
O valor de **n** é muito grande e, portanto, o primeiro termo da inequação nunca será inferior ao segundo termo, para qualquer $k_2 > 0$ e $n \geq n_0$, sendo $n_0 \geq 1$.

Logo, **$2n^2 - 3n + 4$** NÃO é **O(n)**

Notação Big-Oh

Observações

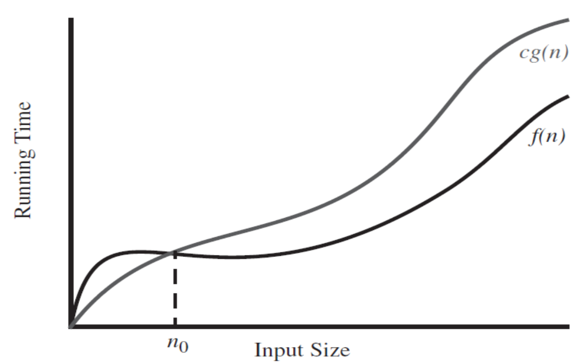
- A ordem de complexidade **O(n)** é melhor que **$O(n^2)$** ou **$O(n^3)$**;
- Embora seja verdade dizer **que $f(n) = 4n^3 + 3n^{4/3}$** seja **$O(n^5)$**, é mais informativo e prático dizer que seja **$O(n^3)$**;
- Algumas funções frequentemente aparecem na Análise de Algoritmos, tais como: **O(log n)**, **O(n)**, **$O(n^2)$**, **$O(n^3)$**, **$O(n^k)$** $(K \geq 1)$ e $O(n^a)$ $(a > 1)$.



Notação Big-Oh

Observações

- É desaconselhável dizer que **f(n) ≤ g(n)**, uma vez que o conceito de **Big-Oh** já denota a desigualdade “menor ou igual”;
- Assim, embora comumente usado, não é completamente correto dizer-se que **f(n) = g(n)**;
- É melhor dizer-se que **f(n) ϵ g(n)**, uma vez que **Big-Oh** denota uma coleção de funções.



The function $f(n)$ is $O(g(n))$, for $f(n) \leq c \cdot g(n)$ when $n \geq n_0$.

Notações Assintóticas

Pergunta

- O que é Limite Superior?

Limite Superior

Resposta

- A notação **O(n)** é utilizada para indicar Limites Superiores para Problemas;
- Dado um problema, por exemplo, o de multiplicação de duas matrizes quadradas de ordem **n** (**$n_x n$**);
- Conhece-se um algoritmo para se resolver este problema, pelo método trivial, de complexidade **$O(n^3)$**.

Limite Superior

Resposta

- Assim, sabe-se que a ordem de complexidade deste problema, ou seja, multiplicação de matrizes quadradas de ordem **n**, não deve superar **O(n^3)**, uma vez que existe um algoritmo que o resolve com esta complexidade;

- Portanto, diz-se que uma **COTA SUPERIOR** ou **LIMITE SUPERIOR** para este problema é **O(n^3)**;

- A **cota superior** de um problema pode mudar se alguém descobrir um outro algoritmo melhor.

Limite Superior

Curiosidades

- Por exemplo, Volker Strassen, matemático alemão, apresentou, em 1969, um algoritmo para Multiplicação de Matrizes Quadradas com Complexidade:

$$O(n^{\log 7}) = O(n^{2.807})$$

- Assim, a cota superior ou limite superior para o problema de multiplicação de matrizes passou a ser $O(n^{2.807})$.

Limite Superior

Curiosidades

- Em 1990, Don Coppersmith e Shmuel Winograd melhoraram esta marca para = O(n $^{2.376}$);
- Em 2010, Andrew James Stothers apresentou um algoritmo de complexidade O(n $^{2.373}$);
- Em 2011, Virginia Vassilevska Williams melhorou ainda mais a cota superior do algoritmo, com uma complexidade O(n $^{2.372}$);
- Portanto, a Cota Superior atual para o problema de multiplicação de matrizes é complexidade O(n $^{2.372}$);
- Quem de vocês, meus alunos, poderá melhorá-la???

Notações Assintóticas

<u>Pergunta</u>

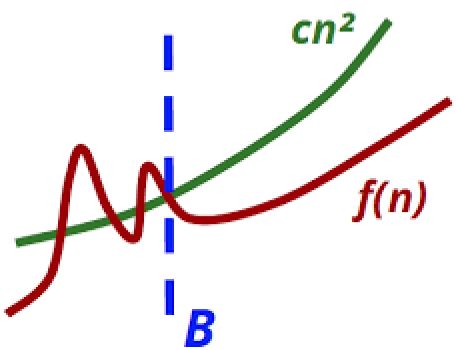
- Existem outras notações além do Big-Oh?

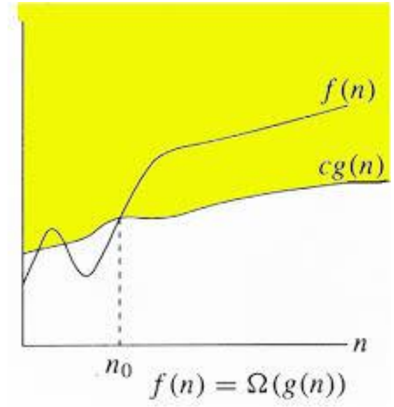
Outras Notações

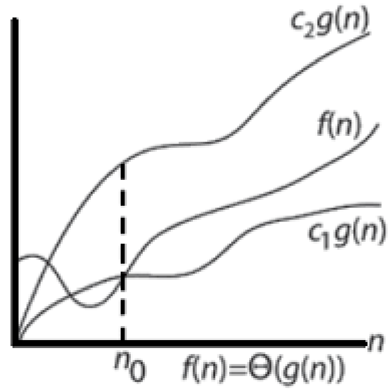
Resposta

- Da mesma forma que a notação **Big-Oh** provê uma forma assintótica de dizer que uma função é “menor ou igual” a outra função, há outras notações que provêm formas assintóticas para fazer outras formas de comparação;

- Em Análise de Algoritmos essas outras formas assintóticas são conhecidas por **Big-Omega** e **Big-Theta**.







Notações Assintóticas

Exercícios

1. Um algoritmo apresenta Função de Complexidade:

$$\mathbf{F(n) = 5n^2 + 10n + 8}$$

Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(n²)**.

Notações Assintóticas

Exercícios

2. Um algoritmo apresenta Função de Complexidade:

$$\mathbf{F(n) = n^3}$$

Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade $\mathbf{O(n^3)}$.

Notações Assintóticas

Exercícios

3. Um algoritmo apresenta Função de Complexidade:

$$F(n) = 5n^2 + 10n + 8$$

Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(n)**.

Notações Assintóticas

Exercícios

4. Um algoritmo apresenta Função de Complexidade:

$$\mathbf{F(n) = n^3}$$

Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(n²)**.

Notações Assintóticas

Exercícios

5. Um algoritmo apresenta Função de Complexidade:

$$F(n) = n$$

Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(n)**.

Notações Assintóticas

Exercícios

6. Um algoritmo apresenta Função de Complexidade:

$$F(n) = 50n^2$$

Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(n)**.

Notações Assintóticas

Exercícios

7. Um algoritmo apresenta Função de Complexidade:

$$F(n) = 50n^2$$

Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(n^3)**.

Notações Assintóticas

Exercícios

8. Um algoritmo apresenta Função de Complexidade:

$$F(n) = n^2 - 200n - 300$$

Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(n)**.

Notações Assintóticas

Exercícios

9. Um algoritmo apresenta Função de Complexidade:

$$F(n) = n^2 - 200n - 300$$

Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade $\mathbf{O(n^5)}$.

Notações Assintóticas

Exercícios

10. Um algoritmo apresenta Função de Complexidade:

$$F(n) = n^2 - 200n - 300$$

Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(1)**.

Notações Assintóticas

Exercícios

11. Um algoritmo apresenta Função de Complexidade:

$$F(n) = 3/2n^2 + 7/2n - 4$$

Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(n)**.

Notações Assintóticas

Exercícios

12. Um algoritmo apresenta Função de Complexidade:

$$F(n) = 3/2n^2 + 7/2n - 4$$

Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(n^2)**.

Notações Assintóticas

Exercícios

13. Um algoritmo apresenta Função de Complexidade:

$$\mathbf{F(n) = 5n^2 + 10n - 900}$$

Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade $\mathbf{O(n^2)}$.

Notações Assintóticas

Exercícios

14. Um algoritmo apresenta Função de Complexidade:

$$F(n) = 5n^2 + 10n - 900$$

Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(n)**.

Notações Assintóticas

Exercícios

15. Um algoritmo apresenta Função de Complexidade:

$$F(n) = 10 + 2/n$$

Prove ou disprove que a Função de Complexidade **F(n)** pertence à ordem de complexidade **O(n^2)**.

Referências bibliográficas

- CORMEN, T.H. et al. Algoritmos: Teoria e Prática (Caps. 13). Campus. 2002.
- ZIVIANI, N. Projeto de algoritmos: com implementações em Pascal e C (Cap. 1). 2.ed. Thomson, 2004.
- FEOFILOFF, P. Minicurso de Análise de Algoritmos, 2010. Disponível em: http://www.ime.usp.br/~pf/livrinho-AA/
- DOWNEY, A.B. *Analysis of algorithms* (Cap. 2), Em: *Computational Modeling and Complexity Science.* Disponível em: http://www.greenteapress.com/compmod/html/book003.html
- ROSA, J.L. Notas de Aula de Introdução a Ciência de Computação II. Universidade de São Paulo. Disponível em: http://coteia.icmc.usp.br/mostra.php?ident=639

Referências bibliográficas

- GOODRICH, Michael T. et al: *Algorithm Design and* Applications. Wiley, 2015.
- LEVITIN, Anany. *Introduction to the Design and Analysis of Algorithms*. Pearson, 2012.
- SKIENA, Steven S. *The Algorithm Design Manual.* Springer, 2008.
- Série de Livros Didáticos. *Complexidade de Algoritmos.* UFRGS.
- BHASIN, Harsh. *Algorithms – Design and Analysis.* Oxford University Press, 2015.
- FREITAS, Aparecido V. de – 2022 – Estruturas de Dados: Notas de Aula.
- CALVETTI, Robson - 2015 – Estruturas de Dados: Notas de Aula.

Aula 06

FIM