Aula 2 - Eficiência Computacional - Parte II

Luís Felipe

UFF

31 de Agosto de 2023

Luís Felipe
31/08/23

Notação O, Ω, Θ

- Ao analisarmos a complexidade de um algoritmo, estamos interessados apenas nos valores assintóticos e eliminamos constantes multiplicativas e aditivas.
  - Dada uma expressão que represente tal complexidade, consideraremos apenas o termo de maior ordem (desprezando a constante).
- Exemplo:
  - $6n^3 \longrightarrow n^3$
  - $n^2 + 5 \longrightarrow n^2$
  - $10n^4 + 4 \longrightarrow n^4$
  - $3n^2 + 5n - 7 \longrightarrow n^2$
  - $2n^3 + \log n \longrightarrow n^3$
- O objetivo é, usando as notações O, Ω e Θ representar situações semelhantes a estas.

Luís Felipe
31/08/23

Notação O

- Desejamos determinar uma função que limite superiormente valores assintóticos da função que queremos representar.

Formalmente,

- Sejam f, h funções reais de variável inteira n. Dizemos que f é $O(h)$, denotado por $f = O(h)$, se existirem uma constante $c > 0$ e um inteiro n_0, tais que:

$$n > n_0 \to f(n) \leq c \cdot h(n)$$

Luís Felipe
31/08/23

Notação O - Exemplo 1

- $f(n) = n - 1$ $h(n) = ?$

  - Observe que $h(n) = n$ é uma função válida, pois, para $c = 1, n_0 = 1, f(n) \leq c.h(n)$.
  - Note que $h(n) = n^2$ também é uma função válida para os mesmos c e n_0, pois, nestes casos, $f(n) \leq c.h(n)$.
  - Da mesma forma, $h(n) = 2^n$ também é válida.

Luís Felipe
31/08/23

Notação O - Exemplo 2

- $f(n) = 10n$ $h(n) = ?$

  - Observe que $h(n) = n$ NÃO é uma função válida para $c = 1, n_0 = 1$.
    - Quando $c = 1$, $f(n) \geq c.h(n)$ para todo $n > n_0$.

O que fazer??

Que c poderá nos salvar?

  - Para $c = 10, n_0 = 1, f(n) \leq ch(n), \forall n > n_0$.
  - Logo, $f = O(h) = O(n)$.
    OBS.: Para $c < 10$, $h(n)$ não limita superiormente $f(n), \ \forall n \geq 1$.
  - Note que $h'(n) = n^2$ também é uma função válida para os mesmos c e n_0, pois, neste caso, $f(n) \leq h'(n)$.
    E se $c = 1$? É possível que $h'(n)$ limite superiormente $f(n)$?

sim $n_0 = 9$

Luís Felipe
31/08/23

Notação O - Exemplo 3

- $f(n) = (n+1)^2$ $h(n) = ?$
  - Se $h(n) = n^2$, quais valores c e n_0 devem assumir para que $f(n) \leq c \cdot h(n), \forall n > n_0$?
  - Note que, quando $c = 1$, $f(n) > h(n), \forall n \geq 0$.
  - Entretanto, quando $c = 4$, $n_0 = 1$, temos $f(n) \leq 4 \cdot h(n), \forall n \geq n_0$.
    Pois, $(n+1)^2 = n^2 + 2n + 1 \leq n^2 + 2n^2 + n^2 = 4n^2$
  - Logo, $f = O(h)$.

Luís Felipe
31/08/23

Complexidade

Exemplo: Complexidade dos algoritmos vistos na última aula

- inversão de sequência: $O(n)$
- cálculo do fatorial (iterativo): $O(n)$
- cálculo do fatorial (recursivo): $O(n)$
- soma de matrizes : $O(n^2)$
- multiplicação de matrizes: $O(n^3)$

Luís Felipe
31/08/23

Notação O - Propriedades

Sejam g, h funções reais positivas e k uma constante.

- $O(g+h) = O(g) + O(h)$
- $O(g \cdot h) = O(g) \cdot O(h)$
- $O(k \cdot g) = k \cdot O(g) = O(g)$

Luís Felipe
31/08/23

Notação Θ

- Exprime limite superior justo

Formalmente,

Sejam f, h funções reais positivas da variável inteira n.
Dizemos que f é $\Theta(h)$, e denotamos por $f = \Theta(h)$, se
$f = O(h)$ e $h = O(f)$

- Ou seja, as duas funções têm a mesma ordem de grandeza assintótica.
- Por ser justa, é preferível utilizar a notação Θ à notação O, sempre que possível.

Luís Felipe

31/08/23

Notação Θ - Exemplo 1

Sejam $f = n^2 - 1$, $h = n^2$.

- $f = O(h)$ $c = 1, n_0 = 1$
- $h = O(f)$ $c = 2, n_0 = 1$
- Logo, $f = \Theta(h)$

Seja $h = n^3$.

- $f = O(h)$ $c = 1, n_0 = 1$
- h NÃO é $O(f)$
- Logo, f não é $\Theta(h)$

Luís Felipe
31/08/23

Notação Θ - Exemplo 2

Sejam $f = 5 + \log n^2 + \log^2 n$, $h = n$.

- f = O(h)
- h NÃO é O(f)
- Logo, f não é $\Theta(h)$

Seja $h = \log^2 n$.

- f = O(h)
- h= O(f)
- Logo, $f = \Theta(h)$

Luís Felipe
31/08/23

Notação Ω

- Exprime limites inferiores

  Formalmente,

  Sejam f, h funções reais positivas da variável inteira n. Dizemos que f é $\Omega(h)$, e denotamos por $f = \Omega(h)$, se existir uma constante $c > 0$ e um valor n_0 tal que

$$n > n_0 \rightarrow f \geq c \cdot h(n)$$

- Em geral, para a notação $\Omega(f)$, procura-se o maior valor possível para f

Luís Felipe

Notação Ω - Exemplo

- $f = n^2 - 1$
  - $f = \Omega(1)$
  - $f = \Omega(n)$
  - $f = \Omega(n^2)$
  - f NÃO é $\Omega(n^3)$

Luís Felipe
31/08/23

Algoritmos Ótimos

- Seja P um problema.
- Um limite inferior para P é uma função ℓ tal que a complexidade de pior caso de qualquer algoritmo que resolva P seja $\Omega(\ell)$.
- Em outras palavras, qualquer algoritmo que resolva P efetua pelo menos $\Omega(\ell)$ passos.
- Um algoritmo para P é ótimo quando sua complexidade de pior caso é igual ao limite inferior de P.

Luís Felipe
31/08/23

OBS. Limites Inferiores

- No contexto de Algoritmos Ótimos, o conceito de limite inferior é relativo a ao problema que quer-se resolver.
- Um limite inferior natural para qualquer problema é o tamanho da entrada quando precisa ser processada.
- Determinar limites inferiores não triviais de problemas é, em geral, uma tarefa árdua.
- Baseia-se no desenvolvimento de propriedades matemáticas inerentes aos problemas e não aos algoritmos utilizados.
- Assim como a notação O é conveniente para exprimir pior caso, a notação Ω é usada para limites inferiores.

Luís Felipe
31/08/23

Algoritmos Ótimos

Exemplos: (algoritmos vistos na aula 1)

1. Inversão de sequências:
   - 1.1 limite inferior natural: $\Omega(n)$
   - 1.2 complexidade de pior caso: $O(n)$

   ótimo

2. Adição de matrizes:
   - 2.1 limite inferior natural: $\Omega(n^2)$
   - 2.2 complexidade de pior caso: $O(n^2)$

   ótimo

3. Multiplicação de matrizes:
   - 3.1 limite inferior natural: $\Omega(n^2)$
   - 3.2 complexidade de pior caso: $O(n^3)$

   ~~ótimo~~ ?