AULA 04 – NOTAÇÃO ASSINTÓTICA

Prof. Daniel Kikuti

Universidade Estadual de Maringá

30 de julho de 2014

Resumo da aula anterior

- Técnica de Projeto de Algoritmos: **Divisão e conquista**.
  - Três passos fundamentais: **dividir**, **conquistar** e **combinar**.

Resumo da aula anterior

- Técnica de Projeto de Algoritmos: **Divisão e conquista**.
  - Três passos fundamentais: **dividir**, **conquistar** e **combinar**.
- Análise de complexidade.
  - Uso de uma **equação de recorrência** $T(n)$ representando o tempo de execução de um problema de tamanho n.

$$T(n) = \begin{cases} c & \text{caso base} \\ aT(n/b) + D(n) + C(n) & \text{caso contrário} \end{cases}$$

Resumo da aula anterior

- Técnica de Projeto de Algoritmos: **Divisão e conquista**.
  - Três passos fundamentais: **dividir**, **conquistar** e **combinar**.
- Análise de complexidade.
  - Uso de uma **equação de recorrência** $T(n)$ representando o tempo de execução de um problema de tamanho n.

$$T(n) = \begin{cases} c & \text{caso base} \\ aT(n/b) + D(n) + C(n) & \text{caso contrário} \end{cases}$$

- Análise do Merge-Sort.

$$T(n) = \begin{cases} c & \text{se } n = 1, \\ 2T(n/2) + cn & \text{se } n > 1. \end{cases}$$

Resumo da aula anterior

- Técnica de Projeto de Algoritmos: **Divisão e conquista**.
  - Três passos fundamentais: **dividir**, **conquistar** e **combinar**.
- Análise de complexidade.
  - Uso de uma **equação de recorrência** $T(n)$ representando o tempo de execução de um problema de tamanho n.

$$T(n) = \begin{cases} c & \text{caso base} \\ aT(n/b) + D(n) + C(n) & \text{caso contrário} \end{cases}$$

- Análise do Merge-Sort.

$$T(n) = \begin{cases} c & \text{se } n = 1, \\ 2T(n/2) + cn & \text{se } n > 1. \end{cases}$$

- Árvore de recorrência.

Conteúdo desta aula

- Introdução a notação assintótica.
- Notação Θ.
- Notação O.
- Notação Ω.
- Outras notações (o e ω).
- Propriedades de comparações assintóticas.
- Notação assintótica em equações.
- Exercícios.

Introdução

As notações que veremos descrevem comportamento de funções.

A análise assintótica de um algoritmo requer:

- Identificar **qual aspecto** analisar:
  - tempo de execução;
  - uso de espaço (quantidade de memória);
  - outros atributos como consumo de banda (comunicação), I/O, etc.;
- Identificar uma **função que caracteriza** tal aspecto.
- Identificar a **classe assintótica de funções** a qual esta função pertence, onde as classes são definidas em termos de limites na taxa de crescimento.

Introdução

Ordem de crescimento

A **ordem de crescimento** caracteriza a eficiência e permite a comparação de desempenho entre diferentes algoritmos.

- Para entradas grandes o suficiente, constantes multiplicativas e termos de baixa-ordem em uma análise exata são dominados pelo efeito do tamanho da entrada.

Introdução

Ordem de crescimento

A **ordem de crescimento** caracteriza a eficiência e permite a comparação de desempenho entre diferentes algoritmos.

- Para entradas grandes o suficiente, constantes multiplicativas e termos de baixa-ordem em uma análise exata são dominados pelo efeito do tamanho da entrada.

Eficiência assintótica

Quando olhamos para entradas suficentemente grandes e consideramos relevante apenas a ordem de crescimento, estamos estudando a eficiência **assintótica** do algoritmo.

- Preocupa-se com como o tempo de execução de um algoritmo aumenta à medida que a entrada aumenta sem limitações.
- Um algoritmo assintoticamente mais eficiente será a melhor escolha em todos os casos, exceto para entradas muito pequenas.

Notação O (O grande – *Big-oh*)

Definição

Dada uma função $g(n)$, denotamos por $O(g(n))$ o conjunto de funções

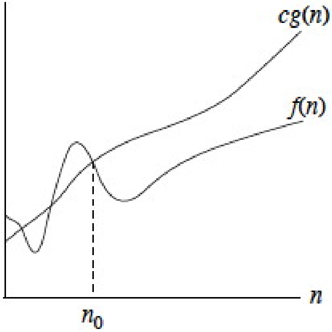
$$\begin{aligned} O(g(n)) &= \{f(n) : \exists \text{ constantes positivas } c \text{ e } n_0 \text{ tais que} \\ &\quad 0 \leq f(n) \leq cg(n), \forall n \geq n_0\} \end{aligned}$$

Notação O (O grande – *Big-oh*)

Definição

Dada uma função $g(n)$, denotamos por $O(g(n))$ o conjunto de funções

$$\begin{aligned} O(g(n)) &= \{f(n) : \exists \text{ constantes positivas } c \text{ e } n_0 \text{ tais que} \\ &\quad 0 \leq f(n) \leq cg(n), \forall n \geq n_0\} \end{aligned}$$



Notação O (O grande – *Big-oh*)

- A notação O descreve um **limite assintótico superior** para uma função (garantia no pior caso).
- Escrevemos $f(n) = O(g(n))$ para indicar que $f(n) \in O(g(n))$.
- Assumimos que todas as funções envolvidas são assintoticamente não negativas.
- Informalmente, dizemos que $f(n)$ cresce no máximo tão rapidamente quanto $g(n)$.

Exemplo:

Mostre que $2n^2 = O(n^2)$.

Notação O (O grande – *Big-oh*)

- A notação O descreve um **limite assintótico superior** para uma função (garantia no pior caso).
- Escrevemos $f(n) = O(g(n))$ para indicar que $f(n) \in O(g(n))$.
- Assumimos que todas as funções envolvidas são assintoticamente não negativas.
- Informalmente, dizemos que $f(n)$ cresce no máximo tão rapidamente quanto $g(n)$.

Exemplo:

Mostre que $2n^2 = O(n^2)$. Considerando a definição, $f(n)$ será $2n^2$ e $g(n)$ será n^2. Temos que mostrar que existe algum c e n_0 tal que $0 \leq 2n^2 \leq cn^2$ para todo $n \geq n_0$.
Isto funciona com $c = 2$, pois isto faz com que os termos f e g sejam equivalentes para todo $n \geq n_0 = 1$.

Notação O (O grande – *Big-oh*)

Mais exemplos:

Pertencem a $O(n^2)$

- n^2
- $n^2 + 1000n$
- $1000n^2 + 1000n$
- $n^{1.99999}$
- n

Notação O (O grande – *Big-oh*)

Mais exemplos:

Pertencem a $O(n^2)$

- n^2
- $n^2 + 1000n$
- $1000n^2 + 1000n$
- $n^{1.99999}$
- n

Não pertencem a $O(n^2)$

- n^3
- $n^{2.000001}$
- $n^2 \lg n$
- 2^n
- $n!$

Exercícios – Big-Oh

Mostre utilizando a definição da notação O, se cada expressão a seguir é verdadeira ou falsa.

1. $100n^2 = O(n^2)$
2. $\frac{1}{2}n^2 - 3n = O(n^2)$
3. $3n + 20 = O(n^2)$
4. $6n^3 = O(n^2)$
5. $720 = O(1)$

Notação O (O grande – *Big-oh*)

Exemplo do Insertion Sort

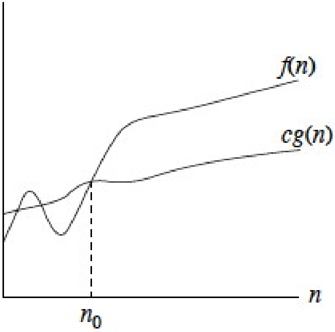
- Verifique que $T(n)$ do Insertion Sort pertence a $O(n^2)$.
- Em geral, um polinômio com termo de maior ordem an^d (polinômio em n de grau d) estará em $O(n^d)$.
- Um limite superior no pior caso também é um limite superior nos outros casos.
- Note que a definição da notação O também funciona para funções $g(n) = n^3$, $g(n) = 2^n$, etc. Então podemos dizer que o pior caso do Insertion Sort é $O(n^3)$, $O(2^n)$, etc. Mas estes limites folgados não são úteis.

Notação Ω (Ômega grande – *Big-omega*)

Definição

Dada uma função $g(n)$, denotamos por $\Omega(g(n))$ o conjunto de funções

$$\Omega(g(n)) = \{f(n) : \exists \text{ constantes positivas } c \text{ e } n_0 \text{ tais que } 0 \leq cg(n) \leq f(n), \forall n \geq n_0\}$$



Notação Ω (Ômega grande – *Big-omega*)

- A notação Ω descreve um **limite assintótico inferior** para uma função de crescimento considerando o melhor caso.
- Informalmente, dizemos que $f(n)$ não cresce tão vagarosamente quanto $g(n)$.

Exemplo:

Mostre que $\sqrt{n} = \Omega(\lg n)$.

Notação Ω (Ômega grande – *Big-omega*)

- A notação Ω descreve um **limite assintótico inferior** para uma função de crescimento considerando o melhor caso.
- Informalmente, dizemos que $f(n)$ não cresce tão vagarosamente quanto $g(n)$.

Exemplo:

Mostre que $\sqrt{n} = \Omega(\lg n)$. Considere $c = 1$ e $n_0 = 16$.

Notação Ω (Ômega grande – *Big-omega*)

Mais exemplos:

Pertencem a $\Omega(n^2)$

- n^2
- $n^2 + 1000n$
- $1000n^2 + 1000n$
- n^3
- $n^{2.000001}$

Notação Ω (Ômega grande – *Big-omega*)

Mais exemplos:

Pertencem a $\Omega(n^2)$

- n^2
- $n^2 + 1000n$
- $1000n^2 + 1000n$
- n^3
- $n^{2.000001}$

Não pertencem a $\Omega(n^2)$

- $n^{1.999999}$
- n
- $\lg n$

Exercícios – Big-Omega

Mostre utilizando a definição da notação Ω, se cada expressão a seguir é verdadeira ou falsa.

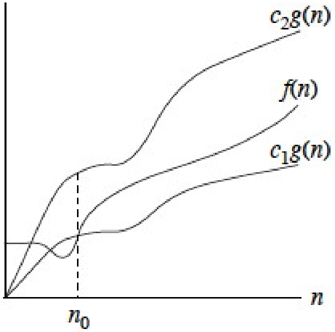
1. $100n^2 = \Omega(n^2)$
2. $\frac{1}{2}n^2 - 3n = \Omega(n^2)$
3. $3n^2 + 20 = \Omega(n)$
4. $6n = \Omega(n^2)$
5. $720 = \Omega(1)$

Notação Θ (Theta)

Definição

Dada uma função $g(n)$, denotamos por $\Theta(g(n))$ o conjunto de funções

$$\begin{aligned}\Theta(g(n)) &= \{f(n) : \exists \text{ constantes positivas } c_1,\ c_2 \text{ e } n_0 \text{ tais que} \\ &\quad 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n),\ \forall n \geq n_0\}\end{aligned}$$



Notação Θ (Theta)

- A notação Θ descreve um **limite assintótico restrito** (justo).
- Informalmente, dizemos que $f(n)$ é "sanduichada" por $c_1 g(n)$ e $c_2 g(n)$.
- Combinação das definições para O e Ω.

Teorema

Para duas funções $f(n)$ e $g(n)$, temos que $f(n) = \Theta(g(n))$ se e somente se $f(n) = O(g(n))$ e $f(n) = \Omega(g(n))$.

Exemplo:

Mostre que $n^2 - 2n = \Theta(n^2)$.

Notação Θ (Theta)

- A notação Θ descreve um **limite assintótico restrito** (justo).
- Informalmente, dizemos que $f(n)$ é "sanduichada" por $c_1 g(n)$ e $c_2 g(n)$.
- Combinação das definições para O e Ω.

Teorema

Para duas funções $f(n)$ e $g(n)$, temos que $f(n) = \Theta(g(n))$ se e somente se $f(n) = O(g(n))$ e $f(n) = \Omega(g(n))$.

Exemplo:

Mostre que $n^2 - 2n = \Theta(n^2)$. Considere $c_1 = 1/2$, $c_2 = 1$ e $n_0 = 4$. Assim, $n^2/2 \leq n^2 - 2 \leq n^2$ para todo $n \geq n_0 = 4$.

Notação Θ (Theta)

Mais exemplos:

Pertencem a $\Theta(n^2)$

- n^2
- $n^2 + 1000n$
- $1000n^2 + 1000n - 2000$

Notação Θ (Theta)

Mais exemplos:

Pertencem a $\Theta(n^2)$

- n^2
- $n^2 + 1000n$
- $1000n^2 + 1000n - 2000$

Não pertencem a $\Theta(n^2)$

- $n^{1.999999}$
- $n^{2.000001}$
- $n \lg n$
- n^3

Exercícios – Θ

Mostre utilizando a definição da notação Θ, se cada expressão a seguir é verdadeira ou falsa.

1. $100n^2 = \Theta(n^2)$
2. $\frac{1}{2}n^2 - 3n = \Theta(n^2)$
3. $3n^2 + 20 = \Theta(n)$
4. $6n = \Theta(n^2)$
5. $720 = \Theta(1)$

Notação o (little-oh)

Definição

Dada uma função $g(n)$, denotamos por $o(g(n))$ o conjunto de funções

$$
\begin{aligned}
o(g(n)) \quad = \quad & \{f(n) : \forall \text{ constante } c > 0, \exists \text{ uma constante } n_0 > 0, \\
& \text{tal que } 0 \leq f(n) < cg(n), \forall n \geq n_0\}
\end{aligned}
$$

- Descreve um limite superior que não é assintoticamente restrito.
- A função $f(n)$ torna-se insignificante em relação a $g(n)$ à medida que n aproxima-se do infinito, isto é

$$\lim_{n \leftarrow \infty} \frac{f(n)}{g(n)} = 0.$$

Notação o (little-oh)

Pertence ou não pertence

- $n^{1.999999} \in o(n^2)$
- $\frac{n^2}{\lg n} \in o(n^2)$
- $n^2 \notin o(n^2)$
- $\frac{n^2}{1000} \notin o(n^2)$

Tente você!

Mostre que $2n^2 = o(n^3)$.

Notação ω (little omega)

Definição

Dada uma função $g(n)$, denotamos por $\omega(g(n))$ o conjunto de funções

$$\begin{aligned}\omega(g(n)) \;=\; & \{f(n) : \forall \text{ constante } c > 0, \exists \text{ uma constante } n_0 > 0, \\ & \text{tal que } 0 \leq cg(n) < f(n), \forall n \geq n_0\}\end{aligned}$$

- Descreve um limite inferior que não é assintoticamente restrito.
- A função $f(n)$ torna-se arbitrariamente grande em relação a $g(n)$ à medida que n aproxima-se do infinito, isto é

$$\lim_{n \leftarrow \infty} \frac{f(n)}{g(n)} = \infty.$$

Notação ω (little-omega)

Pertence ou não pertence

- $n^{2.000001} \in \omega(n^2)$
- $n^2 \lg n \in \omega(n^2)$
- $n^2 \notin \omega(n^2)$

Propriedades de comparações assintóticas

Transitividade

- $f(n) = \Theta(g(n))$ e $g(n) = \Theta(h(n))$ implicam $f(n) = \Theta(h(n))$.
- Vale para as outras notações.

Reflexividade

- $f(n) = \Theta(f(n))$.
- $f(n) = O(f(n))$.
- $f(n) = \Omega(f(n))$.
- E quanto a o e ω?

Simetria

- $f(n) = \Theta(g(n))$ se e somente se $g(n) = \Theta(f(n))$.
- Vale para outras notações? Por quê?

Notação assintótica em equações

Colocando notação assintótica em equações permite-nos simplificar manipulações durante a análise.

Notação assintótica no **lado direito**: $\exists$

$O(g(n))$ no lado direito significa *para alguma* função anônima no conjunto $O(g(n))$.

$$
\begin{array}{ll}
2n^2 + 3n + 1 = 2n^2 + \Theta(n) & \text{significa:} \\
2n^2 + 3n + 1 = 2n^2 + f(n) & \text{para alguma } f(n) \in \Theta(n) \\
 & \text{(em particular, } f(n) = 3n + 1\text{).}
\end{array}
$$

Notação assintótica em equações

Notação assintótica no **lado esquerdo**: $\forall$

A notação somente é usada no lado esquerdo quando também está presente no lado direito.
Semântica: Não importa como a função anônima é escolhida no lado esquerdo, existe um meio de escolher funções no lado direito de forma a satisfazer a equação.
$2n^2 + \Theta(n) = \Theta(n^2)$ significa que $\forall f(n)$, $\exists g(n) \in \Theta(n^2)$ tal que $2n^2 + f(n) = g(n)$.

Combinando Termos

Podemos fazer operações algébricas como:

$$2n^2 + 3n + 1 = 2n^2 + \Theta(n) = \Theta(n^2)$$

Analogia com números reais

- Sejam f, g funções e $a, b \in \mathbb{R}$.
- $f(n) = O(g(n)) \approx a \leq b$.
- $f(n) = \Omega(g(n)) \approx a \geq b$.
- $f(n) = \Theta(g(n)) \approx a = b$.
- $f(n) = o(g(n)) \approx a < b$.
- $f(n) = \omega(g(n)) \approx a > b$.

Exercícios

Livro do Cormen.

- 3.1-1
- 3.1-2
- 3.1-3
- 3.1-4