Aula 2 - Complexidade de Algoritmos

§ Modelo de Máquina

É um conjunto de instruções primitivas que podem ser identificadas num algoritmo. Uma instrução primitiva é uma instrução de baixo nível com um tempo de execução constante. O custo de execução de um algoritmo pode ser medido pelo número de instruções primitivas de um dado modelo de máquina executadas pelo algoritmo.

♠ Exemplo de modelo de máquina:

- atribuição de valores a variáveis
- operações aritméticas
- operações relacionais
- acesso a um arranjo
- retorno do algoritmo

§ Formalização

Seja *a* um algoritmo, e *d* o dado de entrada, definimos :

$\boldsymbol{desemp_a}(d)$ = operações efetuadas na execução de ***a*** com entrada ***d***

temos que,

$\boldsymbol{aval_a}(n) = desemp_a(d)$ [para todo d | tam(d) = n] // um conjunto //

Assim, a **complexidade pessimista** (Cp) de ***a***, para problemas de tamanho ***n***, é:

$\boldsymbol{Cp_a}(n) = \text{Máx}\ \{ aval_a(n) \}$

Algoritmo Ótimo

Sendo *P* um problema, e *K* o conjunto de todos os algoritmos que resolvem *P*, o **algoritmo ótimo** (ao) para *P* é:

$\boldsymbol{ao}(P) = o \in K \mid (\nexists\ c \in K \mid (c \neq o \text{ e } (Cp(c,n) < Cp(o,n))))$

Aplicação

- Complexidade pessimista
- Complexidade média

§ Comportamento Assintótico

É a avaliação do crescimento do resultado de uma função em relação a uma segunda função. É usado na análise da complexidade de algoritmos para valores grandes de entrada da função de complexidade.

Considerando-se f e g : N -> R, define-se os seguintes comportamentos

Cota Assintótica Superior

f(n) é **O** (g(n)) sse $(\exists\, c \in R)\ (\exists\, n_0 \in N)\ (\forall\, n \geq n_0)\ f(n) \leq c.g(n)$

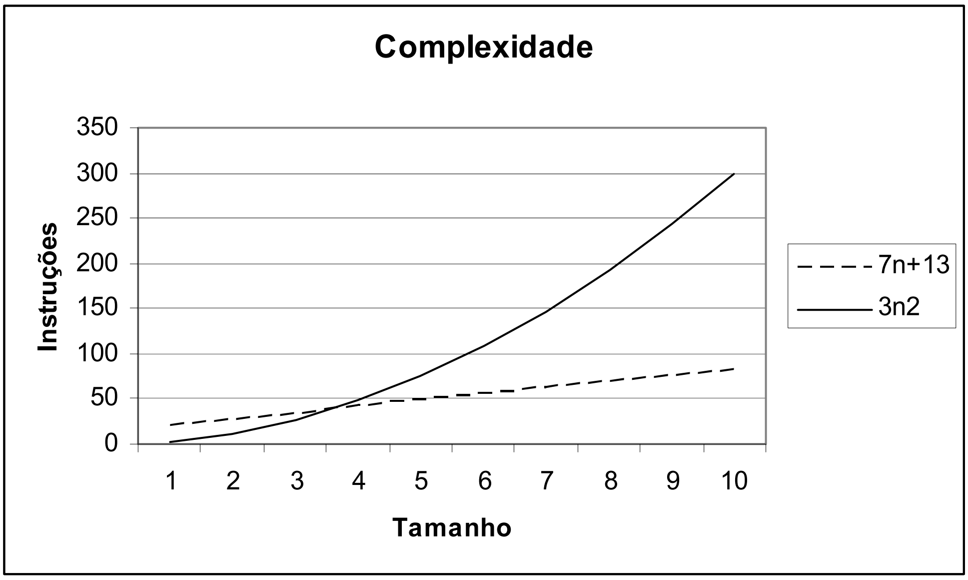
Cota Assintótica Inferior

f(n) é **Ω** (g(n)) sse $(\exists\, d \in R)\ (\exists\, n_0 \in N)\ (\forall\, n \geq n_0)\ g(n) \leq d.f(n)$

Cota Assintótica Exata

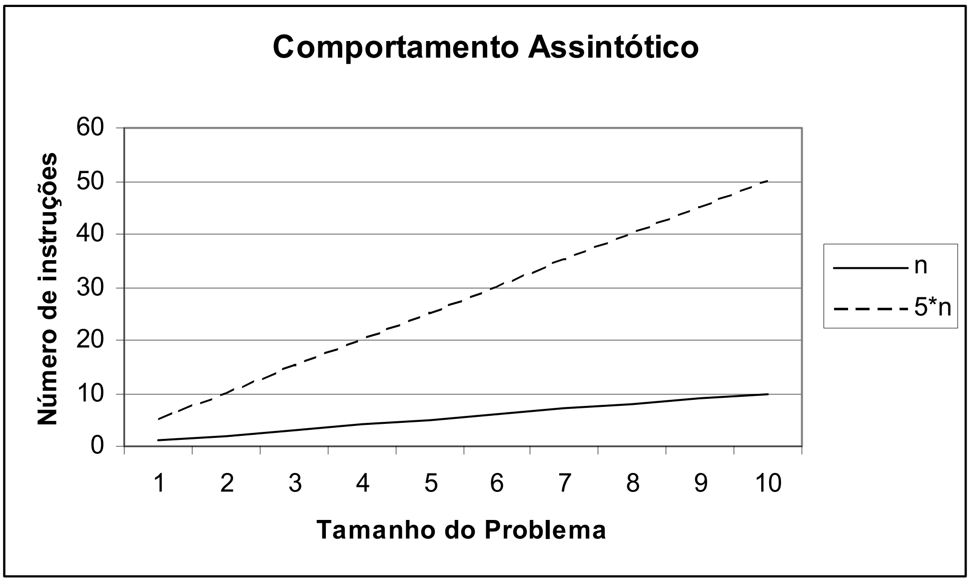
f(n) é **Θ** (g(n)) sse $(\exists\, c, c' \in R)\ (\exists\, n_0 \in N)\ (\forall\, n \geq n_0)\ f(n) \geq c.g(n) \wedge f(n) \leq c'.g(n)$

♠ Exemplo: $f(n) = 7n+13$ e $g(n) = 3n^2$, f é **O**(g)

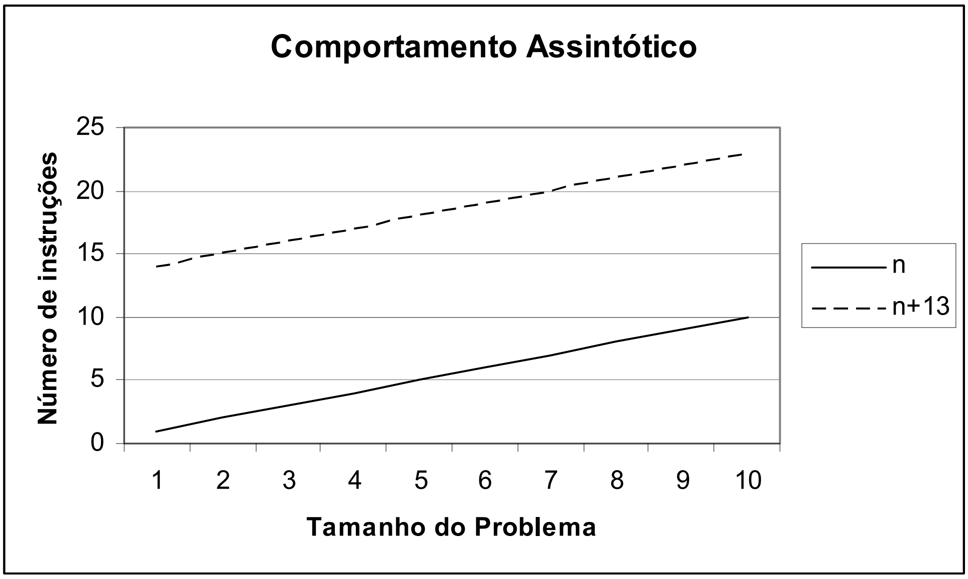


♣ Exercício: demostre que f é O(g) para o exemplo acima.

♠ Exemplo: $f(n) = n$ e $g(n) = 5n$, f é **O**(g)



♠ Exemplo: $f(n) = n$ e $g(n) = n + 13$, f é **O**(g)



♠ Exemplo Complexidade 3n

k.(3n) = t1 , 2 x tamanho -> k.(3(2n)) = t2 -> t2 = 2.t1

§ Regras de Simplificação

Sejam d(n), e(n), f(n) e g(n) funções mapeando inteiros não negativos em reais não negativos, então:

1. Se **d(n)** é O(**f(n)**), então **a.d(n)** é O(**f(n)**) para qualquer constante
2. Se **d(n)** é O(**f(n)**) e **e(n)** é O(**g(n)**), então **d(n)** + **e(n)** é O(**f(n)**+**g(n)**)
3. Se **d(n)** é O(**f(n)**) e **e(n)** é O(**g(n)**), então **d(n)**.**e(n)** é O(**f(n)**.**g(n)**)
4. Se **d(n)** é O(**f(n)**) e **f(n)** é O(**g(n)**), então **d(n)** é O(**g(n)**)
5. Se **f(n)** é um polinômio de grau **d**, então **f(n)** é O($\mathbf{n^d}$)
6. $\mathbf{n^x}$ é O($\mathbf{a^n}$) para quaisquer constantes **x>0** e **a>1**
7. **log($\mathbf{n^x}$)** é O(**log(n)**) para quaisquer constantes **x>0**
8. **(log(n))**$^{\mathbf{x}}$ é O($\mathbf{n^y}$) para quaisquer constantes **x>0** e **y>0**

♠ Exemplo: Simplificar a cota assintótica ($2n^3+4n^2.\log n$) em seu termo mais simples.

Solução: $2n^3+4n^2.\log n$ [r.8] -> $2n^3+4n^2.n$ -> $2n^3+4n^3$ -> $6n^3$ [r.1] -> n^3

Classe de Comportamento Assintótico

Funções que são Θ uma da outra tem o mesmo comportamento Assintótico, formando classes de comportamento assintótico.

Exemplos:

Polinomiais: 1, log n, n, n log n, n^2, n^3

Exponenciais: 2^n , 3^n

♣ Exercícios: Para cada um dos pares de funções abaixo, f e g, indique se f é O(g) e as regras de simplificação que o demonstram.

a) $f(n)=n^2$, $g(n)=n^3.\log_2(n)$