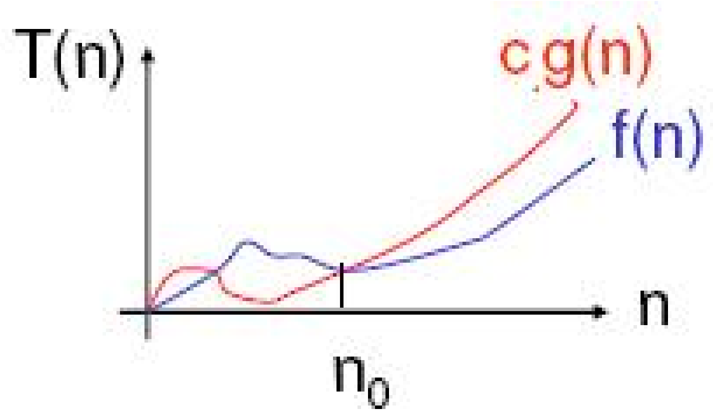
Unidade 5 – Notações Assintóticas



Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPUSP
aparecidovfreitas@gmail.com

Ordem de grandeza de execução

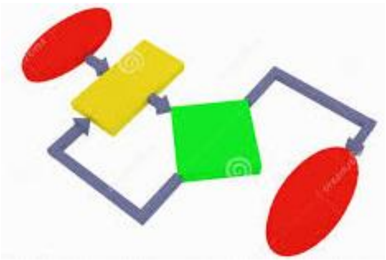
- Por exemplo, o tempo exato de execução de um algoritmo pode ser dado pela função polinomial **f(n) = $3n^2 + 2n + 3$.**

- Neste caso, o tempo aproximado de execução será uma função de **n^2**, ou seja **f(n^2).** (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;

- 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.

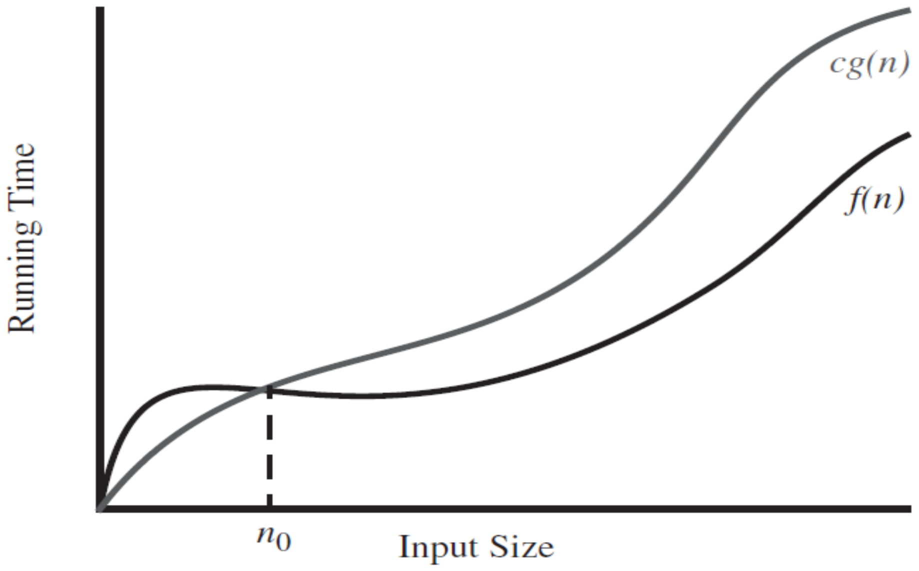


A notação Big-Oh

- 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 $f(n) \leq cg(n)$ para todo inteiro $n \geq n_0$;

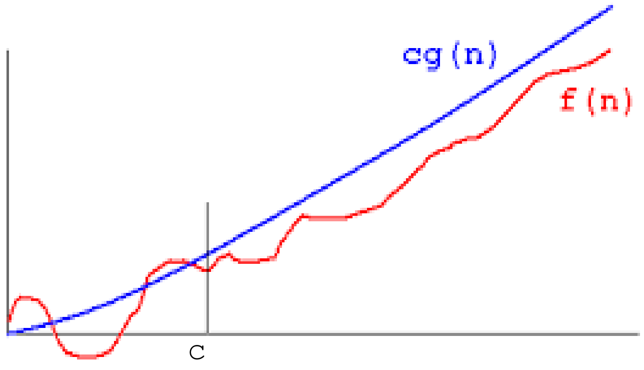
- Essa definição é frequentemente dita **"f(n) é big-Oh de g(n)"** ou **"f(n) é ordem g(n)"**.



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

Ordem de Complexidade O(n)

- A notação **big-Oh** permite que se diga que uma função de n é "menor ou igual" a outra função, por um fator constante (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.



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

Ordem de Complexidade
$F(n) = 3n + 8$

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

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

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

QualitSys

Ordem de Complexidade
$F(n) = 3n + 8$

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_o} \geq$ **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:

Constante c

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

- **Portanto,** $\mathbf{F(n) = 3n + 8n \leq 11.n}$

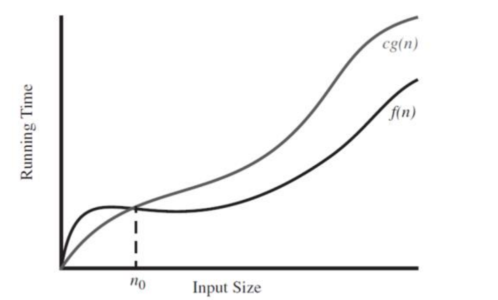
- A expressão acima é verdadeira para todo n > 0;

Ordem de Complexidade
$F(n) = 3n + 8$

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

Constante c

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



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

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

Ordem de Complexidade
$F(n) = 3n + 8$

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

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

Constante c

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

Ordem de Complexidade
F(n) = 3n + 8

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

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

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

- **Portanto, F(n) = 3n + 8 ≤ 11.n**^{2}

Constante c

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

QualitSys

Ordem de Complexidade
F(n) = 3n + 8

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_o} \geq$ **1** tal que **3n + 8 ≤ c n²** para todo inteiro $\mathbf{n} \geq \mathbf{n_0}$.

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

- F(n) = 3n + 8 ≤ **3n² + 8n²**

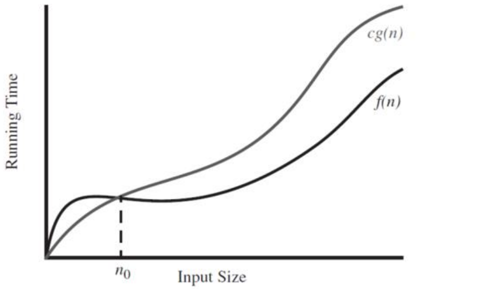
Constante c

- **Portanto, F(n) = 3n + 8 ≤ 11.n²**

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

- Logo, existe **c = 11**, tal que F(n) = 3n + 8 **≤ 11.n², para todo n > 0;**

- **Assim, 3n+8 é O(n²)**



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

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

Ordem de Complexidade
$F(n) = 3n + 8$

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

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

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

Ordem de Complexidade
$F(n) = 3n + 8$

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_o \geq 1}$ 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:

Constante c

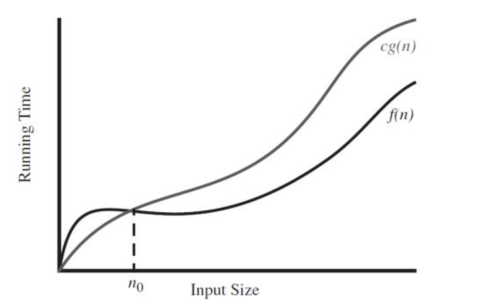
- $F(n) = 3n + 8 \leq \mathbf{3n^3 + 8n^3}$

- **Portanto,** $\mathbf{F(n) = 3n + 8 \leq 11.n^3}$

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

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

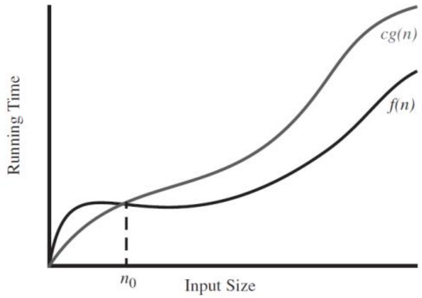
- **Assim, 3n+8 é** $\mathbf{O(n^3)}$



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

Observação

- 3n + 8 é O(**n**).
- 3n + 8 é O(**n²**).
- 3n + 8 é O(**n³**).
- Portanto, **3n+8** ε a um conjunto de funções que atendem à definição de **O(f(n)) ;**



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

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

Ordem de Complexidade
$F(n) = 2n^2 + 3n + 4$ é $O(n^2)$?

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_o \geq 1}$ 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:

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

Ordem de Complexidade
$F(n) = 2n^2 + 3n + 4$ é $O(n^2)$?

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_o \geq 1}$ 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:

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

Constante c

- **Portanto,** $\mathbf{F(n) = 2n^2 + 3n + 4 \leq 9.n^2}$

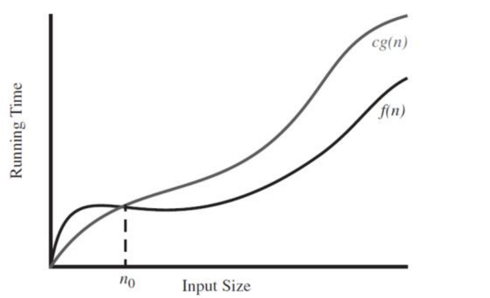
- A expressão acima é verdadeira para todo n > 0;

Ordem de Complexidade
$F(n) = 2n^2 + 3n + 4$ é $O(n^2)$?

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

Constante c

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



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

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

QualitSys

Ordem de Complexidade
$F(n) = 2n^2 + 3n + 4$ é $O(n^3)$?

- Necessita-se de uma constante real **$c > 0$** e uma constante inteira **$n_o \geq 1$** 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:
- $F(n) = \mathbf{2n^2 + 3n + 4 \leq 2n^3 + 3n^3 + 4n^3}$

Constante c

Ordem de Complexidade
$F(n) = 2n^2 + 3n + 4$ é $O(n^3)$?

- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_o \geq 1}$ 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:
- $F(n) = \mathbf{2n^2 + 3n + 4 \leq 2n^3 + 3n^3 + 4n^3}$
- **Portanto,** $\mathbf{F(n) = 2n^2 + 3n + 4 \leq 9.n^3}$
- A expressão acima é verdadeira para todo $n > 0$;

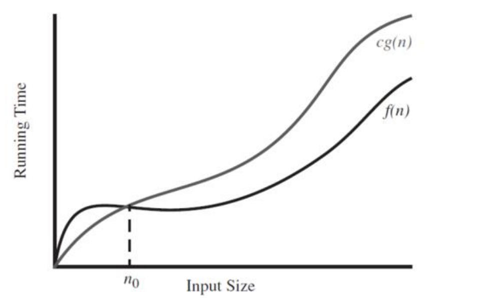
Constante c

Ordem de Complexidade
$F(n) = 2n^2 + 3n + 4$ é $O(n^3)$?

- Necessita-se de uma constante real **c > 0** e uma constante inteira **$n_o \geq 1$** 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:
- $F(n) = \mathbf{2n^2 + 3n + 4 \leq 2n^3 + 3n^3 + 4n^3}$

Constante c

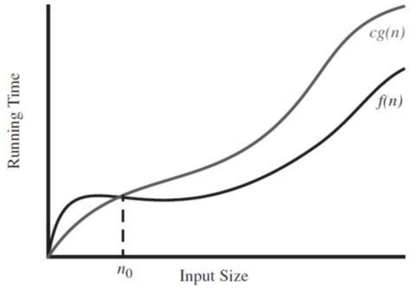
- **Portanto, $F(n) = 2n^2 + 3n + 4 \leq 9.n^3$**
- A expressão acima é verdadeira para todo n > 0;
- Logo, existe **c = 9**, tal que $F(n) =$ **$2n^2 + 3n + 4 \leq 9.n^3$, para todo n > 0;**
- **Assim, $2n^2 + 3n + 4$ é $O(n^3)$**



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

Conclusão

- $2n^2 + 3n + 4$ **é** $O(n^2)$
- $2n^2 + 3n + 4$ **é** $O(n^3)$
- Porém, por razões de ordem **prática**, prefere-se dizer que $2n^2 + 3n + 4$ **é** $O(n^2)$



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

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

Ordem de Complexidade
$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

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

- $\mathbf{2n^2 - 3n + 4 \leq}$ $\mathbf{cn}$

Ordem de Complexidade
$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

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

Ordem de Complexidade
$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

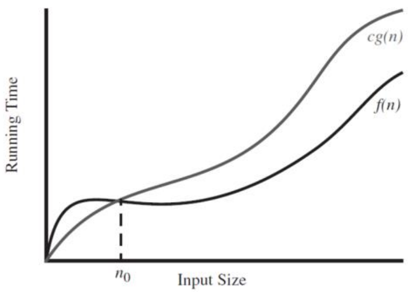
- Necessita-se de uma constante real **$c > 0$** e uma constante inteira **$n_0 \geq 1$** tal que **$2n^2 - 3n + 4 \leq c\,n$** para todo inteiro **$n \geq n_0$** .

- **$2n^2 - 3n + 4 \leq cn$**

- **$2n^2 + 4 \leq cn + 3n$**

- **$2n^2 + 4 \leq n.(c + 3)$** **(c+3) também é uma constante k>0**

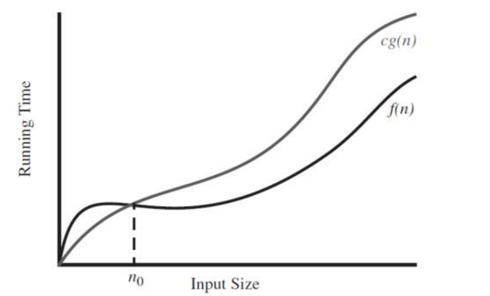
- **$2n^2 + 4 \leq n.k$**



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

Ordem de Complexidade
$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

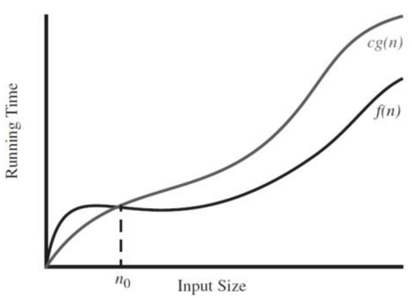
- Necessita-se de uma constante real $\mathbf{c > 0}$ e uma constante inteira $\mathbf{n_o \geq 1}$ tal que $\mathbf{2n^2 - 3n + 4 \leq c\,n}$ para todo inteiro $\mathbf{n \geq n_0}$.
- $\mathbf{2n^2 - 3n + 4 \leq cn}$
- $\mathbf{2n^2 + 4 \leq cn + 3n}$
- $\mathbf{2n^2 + 4 \leq n.(c + 3)}$ **(c+3) também é uma constante k>0**
- $\mathbf{2n^2 + 4 \leq n.k}$
- $\mathbf{2n^2 + 4 \leq n.k}$ **=> ABSURDO!**
- n é muito grande e, portanto, o primeiro termo da inequação nunca será inferior ao segundo termo, para qualquer **k>0** e $n \geq \mathbf{n_0}$ **, sendo** $\mathbf{n_o \geq 1}$
- **Logo,** $2n^2 - 3n + 4$ NÃO é $O(n)$!



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

Ordem de Complexidade
$F(n) = 2n^2 - 3n + 4$ é $O(n)$?

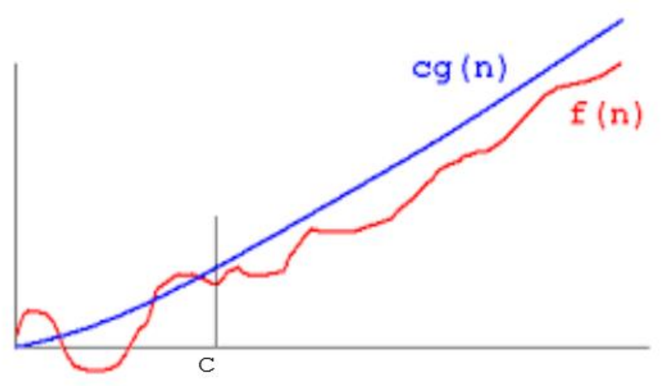
- Necessita-se de uma constante real **c > 0** e uma constante inteira $\mathbf{n_0 \geq 1}$ tal que $\mathbf{2n^2 - 3n + 4 \leq c\,n}$ para todo inteiro $\mathbf{n \geq n_0}$.
- $\mathbf{2n^2 - 3n + 4 \leq cn}$
- $\mathbf{2n^2 + 4 \leq cn + 3n}$
- $\mathbf{2n^2 + 4 \leq n.(c + 3)}$ **(c+3) também é uma constante k>0**
- $\mathbf{2n^2 + 4 \leq n.k}$
- $\mathbf{2n^2 + 4 \leq n.k}$ **=> ABSURDO!**
- n é muito grande e, portanto, o primeiro termo da inequação nunca será inferior ao segundo termo, para qualquer **k>0** e $n \geq \mathbf{n_0}$ **, sendo** $\mathbf{n_0 > 1}$
- **Logo,** $2n^2 - 3n + 4$ NÃO é $O(n)$!



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

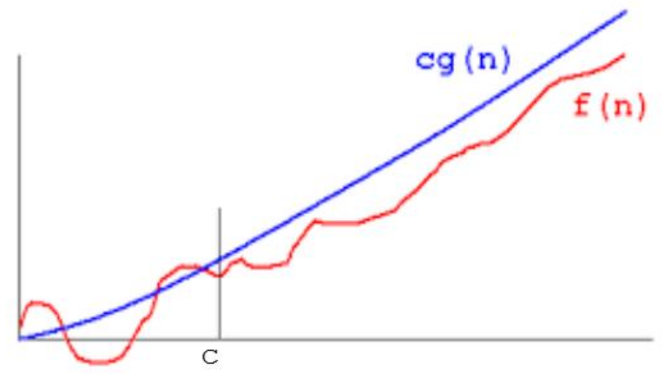
Notação Big Oh - Observações

- 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 $O(n^a)$ $(a > 1)$



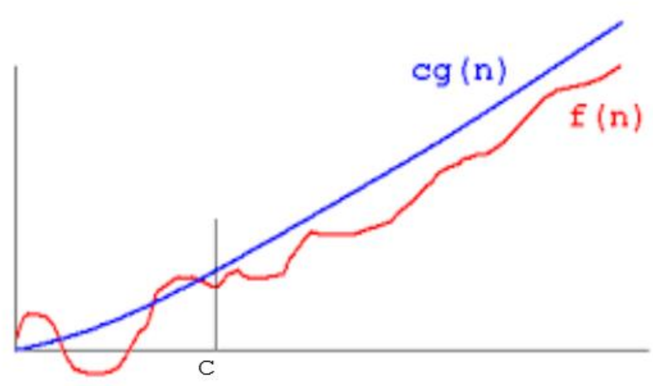
Notação Big Oh – Mais Observações

- 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 **BigOh** denota uma **coleção** de funções.



Limite Superior

- 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 **$O(n^3)$**.

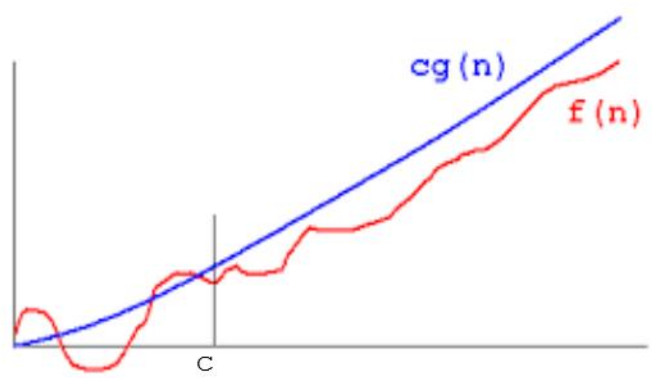


Limite Superior

- Assim, sabe-se que a ordem de complexidade deste problema (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 **$O(n^3)$;**

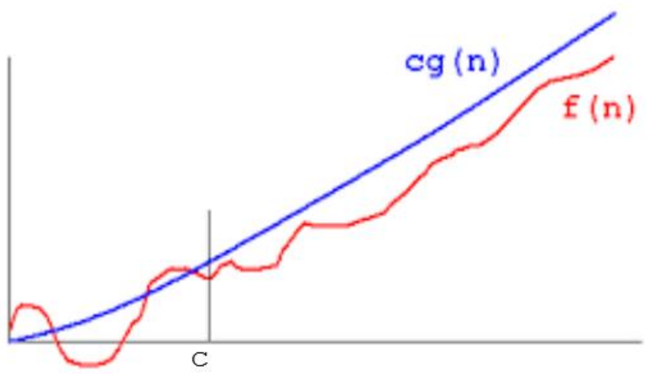
- A **cota superior** de um problema pode mudar se alguém descobrir um outro algoritmo melhor.



Limite Superior

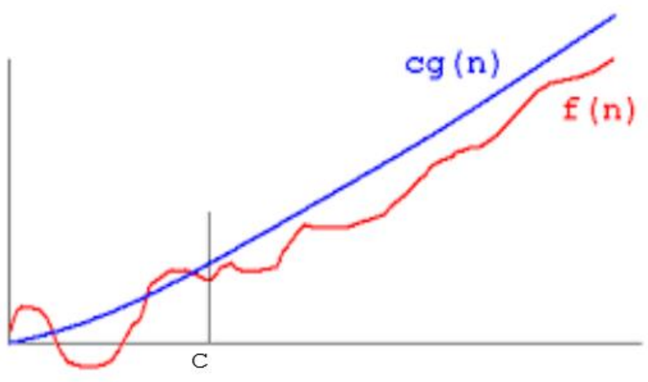
- **V. Strassen** apresentou em **1969** um algoritmo para Multiplicação de Matrizes Quadradas com Complexidade **$O(n^{\log 7}) = 0(n^{2.807})$;**

- Assim, a cota superior ou limite superior para o problema de multiplicação de matrizes passou a ser **$0(n^{2.807})$;**



Limite Superior

- Em **1990, Coppersmith e Winograd** melhoraram esta marca para = **0 (n $^{2.376}$);**
- Em **2010**, A. Stothers apresentou um algoritmo de complexidade **0 (n $^{2.373}$);**
- Em **2011**, V. Willians melhorou ainda mais a cota superior do algoritmo com uma complexidade **0 (n $^{2.372}$);**
- **Portanto, a Cota Superior atual para o problema de multiplicação de matrizes é 0 (n $^{2.372}$);**



Analogia com Record Mundial

- A **cota superior** para um problema é análoga ao **Record Mundial** de uma modalidade de esporte, por exemplo, **Atletismo**;

- Ele é estabelecido pelo melhor atleta (algoritmo) do momento;

- Assim, como o record mundial, a Cota Superior, pode ser melhorada por um algoritmo (atleta) maiz veloz.

Cota Superior – 100m rasos

1998	Carl Lewis	9s92
1993	Linford Christie	9s87
1999	Maurice Greene	9s79
2007	Asata Powel	9s74
2008	Usain Bolt	9s72
2009	Usain Bolt	9s58

Outras Notações

- 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**.