A) A complexidade do algoritmo se baseia em vários aspectos, entre eles, quantidade de trabalho requerido (de super importância), quantidade de espaço, simplicidade, etc. Logo, temos que são fatores tão adversos que apenas um número não irá basear se o trabalho requerido. Pelo contrário, devemos ter em mente que há muitos fatores que caracterizam o trabalho de um algoritmo. Um dos exemplos que é comumente citado são as complexidades de algoritmo. Onde é calculado o tempo de execução baseado pelas instruções executadas, tempo assim uma comparação entre: tempo x tamanho do problema.

B) As ordens Assintóticas são de forma curiosa diferente do pensamento comum. São inversamente proporcionais, entrada relativamente pequenas não são tão eficientes. Portanto são mais eficientes com qualquer entrada excetas as pequenas. Em geral são definidas pelo crescimento da complexidade para entradas suficientemente grandes.

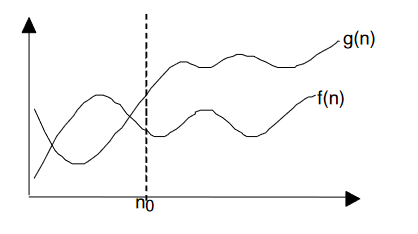
Para definirmos, comecemos por 2 pontos básicos:

1 – As funções *f* e *g* são números reais não negativos (IR+).

2 - Uma *cota assintótica superior* (CAS) é uma função que cresce mais rapidamente do que outra: está acima, a partir de certo ponto.

Comecemos por definir o que significa dada ordem, *g* uma cota assintótica superior para *f*, se:

, representando graficamente:

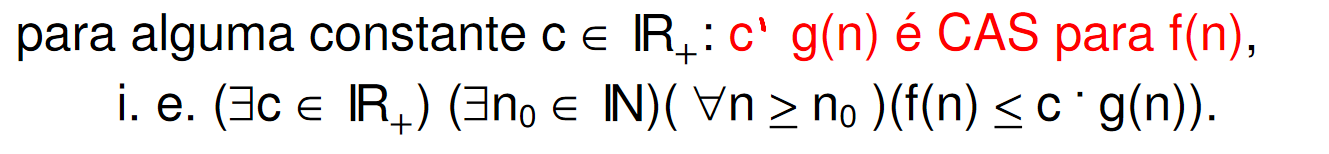


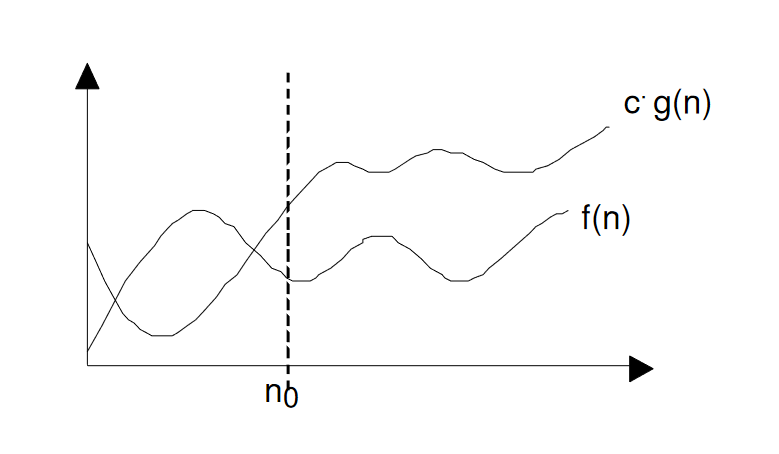
Com isto, chegamos à conclusão de que *g* domina *f*.

**Notações:**

**O**: Cota assintótica superior

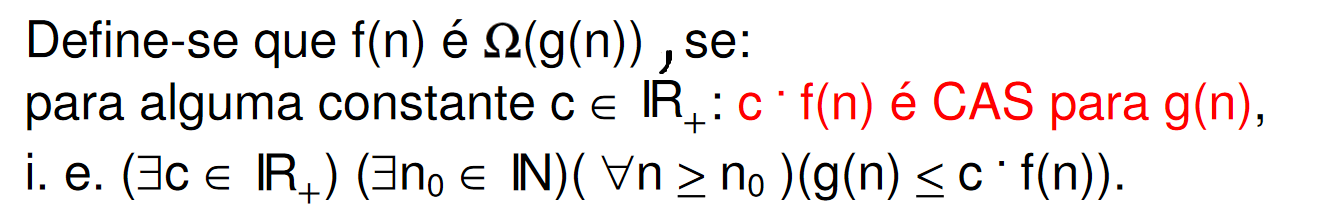


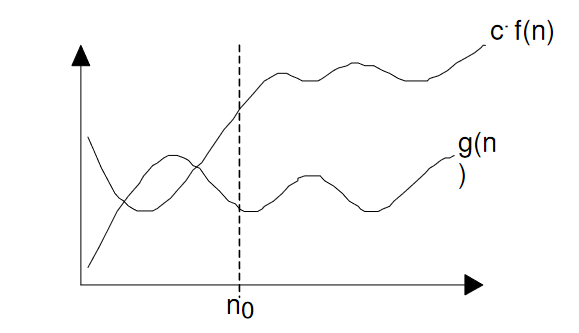




Logo, temos que c’ g(n) é superior a f(n), comprovando assim a notação O.

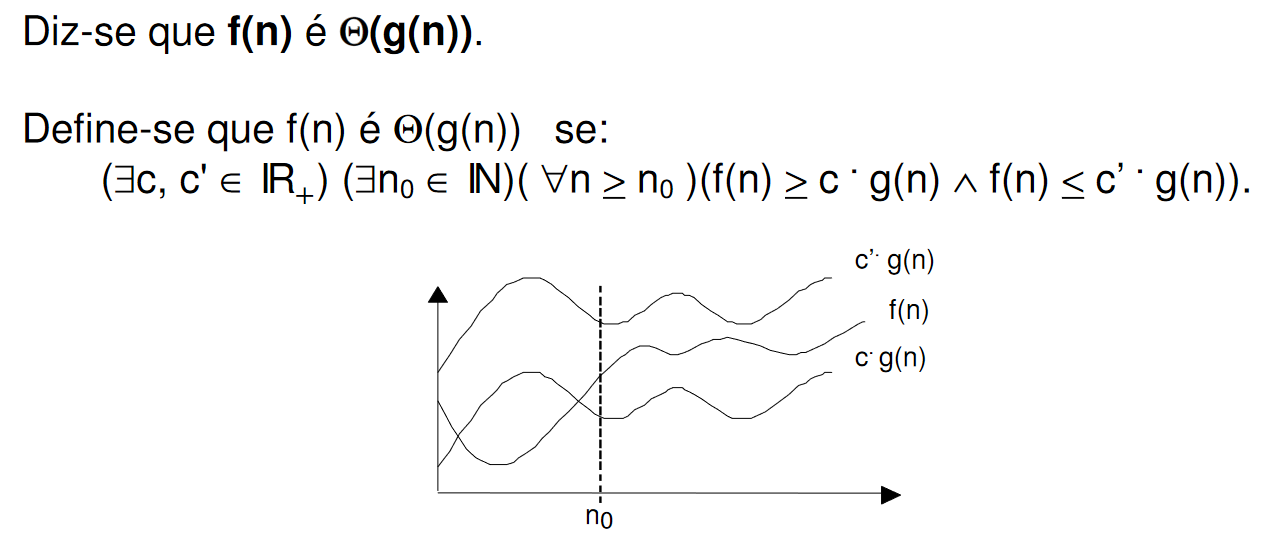
**Ω:** Cota assintótica inferior





Logo, temos que c’ f(n) é superior a f(n), comprovando assim a notação **Ω**.

Ꚛ: limite assintótico exato



Logo, temos que c’ g(n) é superior a f(n), e f(n) inferior a  ,comprovando assim a notação.

C) As classificações de algoritmo podem varia de acordo com o conjunto de esforços para se criar o algoritmo em si. Porém, há medições que podem ser feitas. Estas medições se condensam em (1) execução; (2) custo; (3) desempenho;

A função de desempenho não é constante, fazendo com que, surja 2 critérios importantes a serem notados:

1. Complexidade pessimista (pior caso), quando a complexidade é tomada como máxima para qualquer entrada de um dado “tamanho”.
2. Complexidade média (caso médio), quando é levada em conta a probabilidade de ocorrência de cada entrada.

Dada as complexidades, é importante traçar um paralelo de ambas para que possamos trabalhar dentro de uma escala de previsibilidade.

D) As principais medidas de complexidade são: (1) tempo; (2) espaço;

Ambas as complexidades são auto explicativas, porém, podemos pontuar aspectos interessantes em ambas.

Tempo – É o tempo de execução do algoritmo, pode variar de acordo com cada máquina, compilador, arquitetura, etc. Além disto, é um critério fortemente dependente da implementação.

Espaço – É o espaço alocado dentro da máquina de variáveis, dados, instruções, etc. O espaço pode ser tanto em disco quando em memória volátil.

E) Também chamado de esforço requerido, a quantidade de trabalho por um algoritmo não pode ser descrita simplesmente por um número, porém, podemos denotar o esforço através de complexidades, como a complexidade no pior caso e a complexidade média.

O esforço realizado por um algoritmo é calculado a partir da quantidade de vezes que a operação fundamental é executada. Podemos citar o exemplo de um algoritmo de ordenação, no qual a operação fundamental é a comparação de elementos para pôr em ordem.