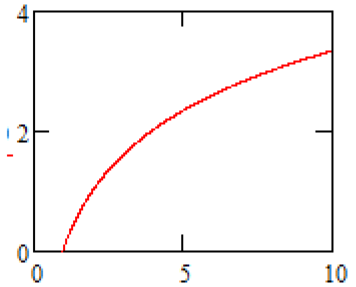
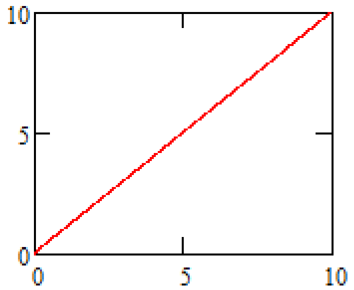
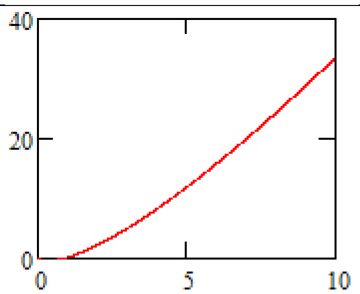
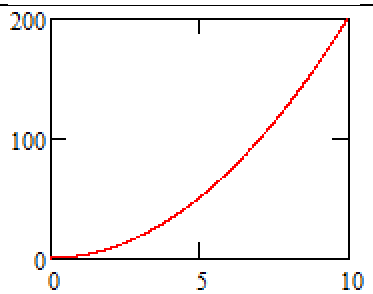
Estudando complexidade de algoritmos

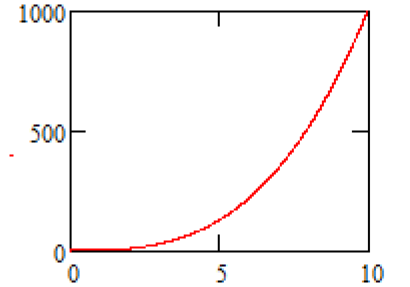
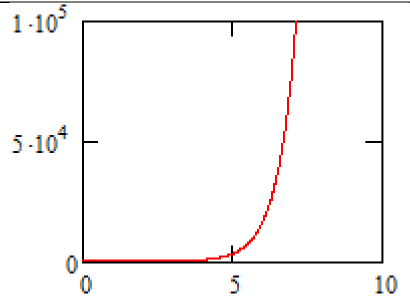
Danilo de Oliveira Domingos – www.dandomin.com.br

Notas de aula de Estrutura de Dados e Análise de Algoritmos (Professor André Balan, mestrado UFABC)

Durante os estudos de complexidade de algoritmos é muito comum se deparar com expressões do tipo "o grau assintótico é linear", "o grau assintótico é quadrático", etc. Então porque não dar uma revisada nas principais curvas envolvidas nessa análise? A tabela abaixo mostra como determinadas funções se comportam no intervalo de domínio (eixo x) entre 0 e 10. Perceba como o intervalo da imagem (eixo y) varia muito.

Tabela 1. Principais funções utilizadas na comparação de complexidade de algoritmos.

Nome	Características	Equação	Gráfico
Logarítmico	Dada a equação, quando n varia de 0 até 10, f(n) varia de 0 até 4.	$f(n) = \log_2 n$	
Linear	Dada a equação, quando n varia de 0 até 10, f(n) varia de 0 até 10.	$f(n) = n$	
Logarítmico vezes Linear	Dada a equação, quando n varia de 0 até 10, f(n) varia de 0 até 40.	$f(n) = n \log_2 n$	
Quadrático	Dada a equação, quando n varia de 0 até 10, f(n) varia de 0 até 200.	$f(n) = 2n^2$	

Cúbico	Dada a equação, quando n varia de 0 até 10, f(n) varia de 0 até 1000.	$f(n) = n^3$	
Exponencial	Dada a equação, quando n varia de 0 até 10, f(n) varia de 0 até 100000.	$f(n) = 5^n$	

Antes de continuar vamos verificar todas juntas num só gráfico:

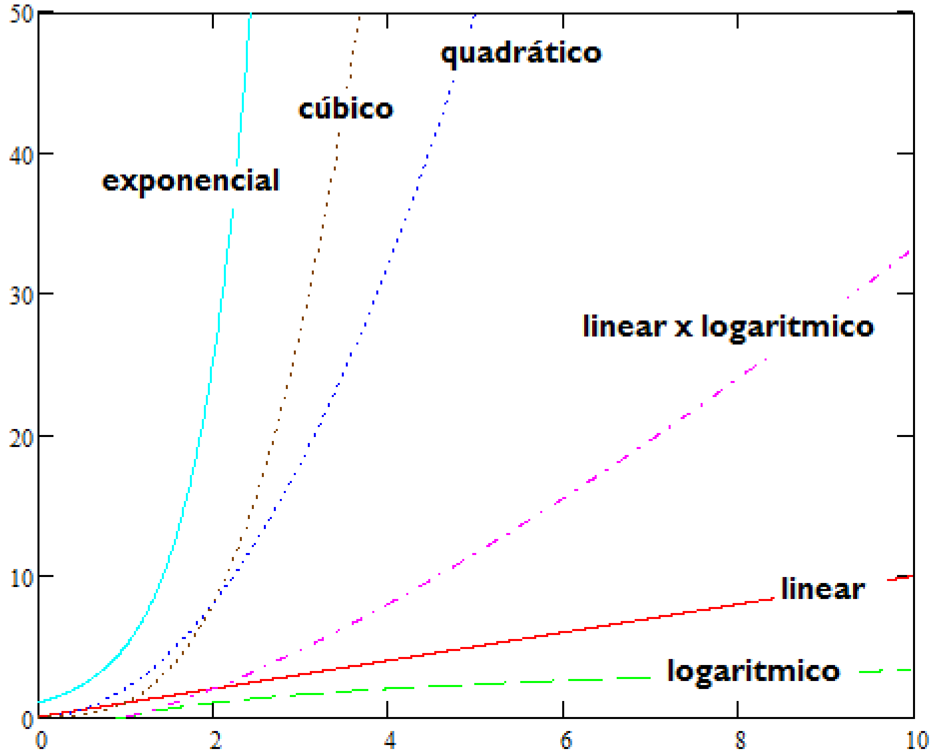


Figura 1. Comparação das curvas no mesmo gráfico. Eixo Y limitado em 50.

Comparando as curvas do gráfico podemos dizer que **se** o seu algoritmo "tem um comportamento" quadrático, no que diz respeito a tempo de processamento em função do número de entradas, **então** ele gasta muito mais tempo que um algoritmo linear. É claro que existe um jeito melhor de dizer o comportamento do algoritmo, que será visto adiante (Ω, Θ, O).

No que diz respeito ao grau assintótico de uma função, apenas o grau da "tendência" já é o suficiente para comparar algoritmos então as constantes envolvidas nas funções tendem a ser tratadas como constantes irrelevantes, por exemplo:

- Grau assintótico Linear: $f(n) = 5n + 9$ ou $f(n) = n$ são tratados como $f(n) = an + b$
- Grau assintótico Logarítmico: $f(n) = 5\log_2 3n$ é tratado como $f(n) = \log n$
- Grau assintótico Quadrático: $f(n) = 3n^2 + 2n + 8$ é tratado como $f(n) = n^2$
- Grau assintótico Cúbico: $f(n) = n^3 + 2n^2 + 5$ é tratado como $f(n) = n^3$
- Grau assintótico Exponencial: $f(n) = 5^n$ ou $f(n) = 2^n$ é tratado como $f(n) = a^n$

Para os exemplos da Figura 1 (e Tabela 1) podemos dizer que:

- Para uma entrada de 10 elementos, o algoritmo mais eficiente seria o que faz suas rotinas seguindo uma tendência logarítmica.
- Para a mesma entrada de 10 elementos, o algoritmo menos eficiente seria o que faz suas rotinas seguindo uma tendência exponencial.
- Mas, o que é um algoritmo bom ou ruim? Um que é mais rápido, porem gasta muito mais memória, ou vice-versa? Isso depende bastante do objetivo e disponibilidade de recursos.

Limites assintóticos

Um algoritmo, dado um conjunto de entrada de dados, pode processá-los de varias formas. No caso de um algoritmo de ordenação, de uma forma grosseira, se os dados já estiverem ordenados então nada precisa ser feito, caso contrário a ordenação é necessária. Quando a análise de complexidade é feita, os dois **casos extremos** são levados em consideração:

- Limite assintótico superior **O**: será uma função que traduz o grau assintótico do tempo **máximo** gasto em função do tamanho da entrada de dados. Exemplo: $O(n)$ ou $O(n^2)$, querem dizer respectivamente, limite assintótico superior linear e limite assintótico superior quadrático. Geralmente é associado ao pior caso de entrada de dados.
- Limite assintótico inferior **Ω**: será uma função que traduz o grau assintótico do tempo **mínimo** gasto em função do tamanho da entrada de dados. Exemplo: $\Omega(n^2)$ ou $\Omega(n\log n)$, querem dizer respectivamente, limite assintótico inferior quadrático e limite assintótico inferior nlogn. Geralmente é associado ao melhor caso de entrada de dados.
- Limite assintótico restrito Θ: em alguns casos, os limites assintóticos superior e inferior terão o **mesmo grau**, neste caso, temos $O(n) = \Omega(n) = \theta(n)$.

Por exemplo, suponha um algoritmo que apresente $O(n^2)$ e $\Omega(n)$, isso significa que o tempo **máximo** que ele demora a realizar sua tarefa tende a uma curva quadrática; da mesma forma que o tempo **mínimo** que ele demora a realizar sua tarefa tende a uma curva linear. A Figura 2 apresenta esta visualização graficamente. Você pode estar se perguntando, mas logo no começinho do gráfico existe um ponto onde os limites assintóticos trocam de lugares. Isto está correto, entre 4 e 2 ocorre uma inversão e para entradas de dados inferiores aquele valor o algoritmo terá $O(n)$ e $\Omega(n^2)$. Mas o objetivo da análise de complexidade é verificar o tempo de processamento do algoritmo em função de **grandes quantidades de dados**, desta forma, o começo do gráfico não terá muita utilidade para 180000 dados de entrada, por exemplo.

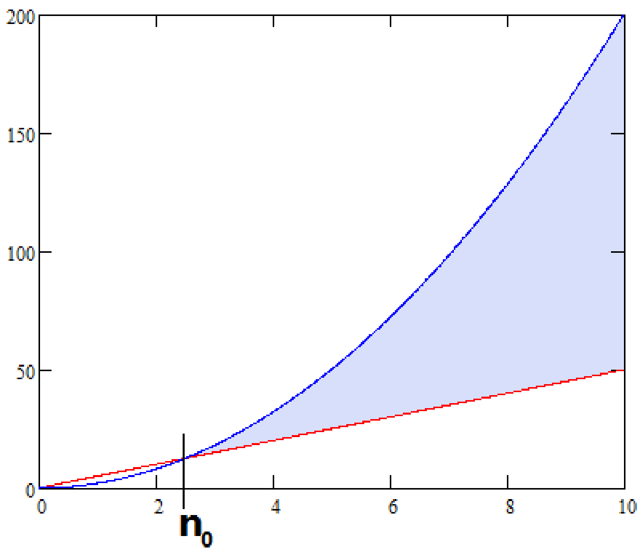


Figura 2. Algoritmo com limites assintóticos superior e inferior quadrático e linear, respectivamente. Estes dois limites correspondem ao melhor caso e ao pior caso, mas a área em azul compreende todos os outros possíveis casos.

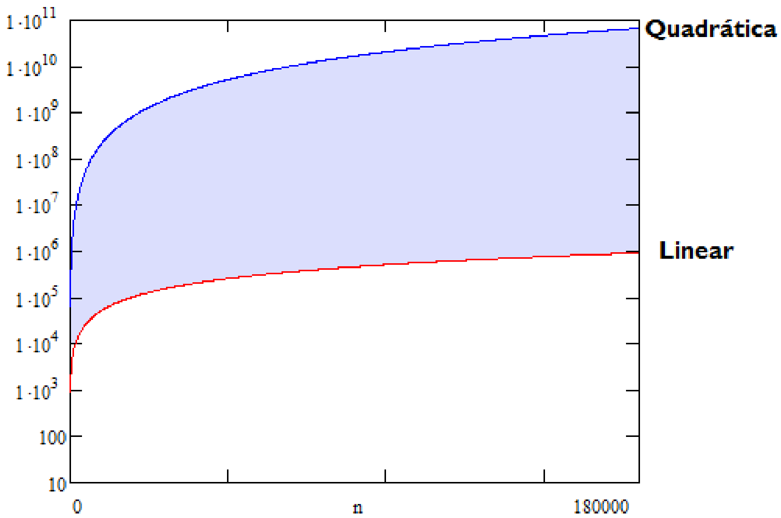


Figura 3. Gráfico para entrada de dados variando de 0 até 180000. Para n = 180000, tempo gasto no pior caso é aproximadamente 100.000 vezes maior que o tempo gasto no melhor caso. Para uma melhor visualização o gráfico é log-linear.

Para provar os limites assintóticos superior, inferior ou restrito, a seguinte demonstração matemática deve ser feita:

- Limite assintótico superior **O**(g(x)).
  Dizer que uma função f(x) tem limite assintótico superior **O**(g(x)) significa dizer:
  Existe uma constante positiva c_1 **e** n_0 **tais que** $0 \leq f(x) \leq c_1 g(x)$ **para** $n \geq n_0$.

  Ou seja, a função f(x) é sempre menor do que a função g(x) multiplicada por uma constante, para $n \geq n_0$
  No exemplo da **Figura 2**, n_0 corresponde àquele ponto entre 2 e 4.

- Limite assintótico restrito **Θ**(g(x)).
  Dizer que uma função f(x) tem limite assintótico restrito Θ(g(x)) significa dizer:
  Existem constantes positivas c_1, c_2 **e** n_0 **tais que** $0 \leq c_1 g(x) \leq f(x) \leq c_2 g(x)$ **para** $n \geq n_0$.

  Ou seja, a função f(x) está sempre entre (restrita) duas funções g(x) iguais a menos de uma constante para $n \geq n_0$.

- Limite assintótico inferior **Ω**(g(x)).
  Dizer que uma função f(x) tem limite assintótico inferior Ω(g(x)) significa dizer:
  Existe uma constante positiva c_1 **e** n_0 **tais que** $0 \leq c_1 g(x) \leq f(x)$ para $n \geq n_0$.

  Ou seja, a função f(x) é sempre maior que g(x) multiplicada por uma constante, para $n \geq n_0$.

Na maioria dos casos, as condições não são respeitadas para todo o domínio de **x**, mas basta provar que a partir de certo n_0 as condições são válidas.

Exemplos:

(dps eu faço!)