Projeto e Análise de Algoritmos

Crescimento assintótico de funções

Atílio G. Luiz

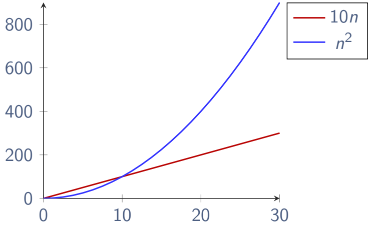
Primeiro Semestre de 2024

Notação assintótica e crescimento de funções

Comportamento assintótico

Para valores pequenos de n, praticamente qualquer algoritmo custa pouco para ser executado, mesmo os ineficientes.

- **Logo:** a escolha do algoritmo tem pouquíssima influência em problemas de tamanho pequeno.



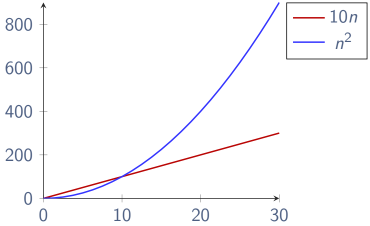
A análise de algoritmos é realizada para valores grandes de n.

- Estudamos o **comportamento assintótico** das funções de complexidade: comportamento da função para valores grandes de n.

Comportamento assintótico

Para valores pequenos de n, praticamente qualquer algoritmo custa pouco para ser executado, mesmo os ineficientes.

- **Logo:** a escolha do algoritmo tem pouquíssima influência em problemas de tamanho pequeno.



A análise de algoritmos é realizada para valores grandes de n.

- Estudamos o **comportamento assintótico** das funções de complexidade: comportamento da função para valores grandes de n.

Notação assintótica

- Expressamos complexidade como uma função em n.
- A função mede o número de instruções de um algoritmo.
- Então, as funções consideradas são sempre não-negativas.
  - Uma função f é **assintoticamente não-negativa** se existe M tal que $f(n) \geq 0$ para todo n maior que M.
- Dependendo do caso, n representa diferentes valores:
  - Problemas de precisão arbitrária: número de bits.
  - Problemas em grafos: número de vértices e/ou arestas
  - Problemas com vetores: tamanho do vetor
  - Problemas de busca em textos: tamanho das strings

Notação assintótica

- Expressamos complexidade como uma função em n.
- A função mede o número de instruções de um algoritmo.
- Então, as funções consideradas são sempre não-negativas.
  - Uma função f é **assintoticamente não-negativa** se existe M tal que $f(n) \geq 0$ para todo n maior que M.
- Dependendo do caso, n representa diferentes valores:
  - Problemas de precisão arbitrária: número de bits.
  - Problemas em grafos: número de vértices e/ou arestas
  - Problemas com vetores: tamanho do vetor
  - Problemas de busca em textos: tamanho das strings

Notação assintótica

- Expressamos complexidade como uma função em n.
- A função mede o número de instruções de um algoritmo.
- Então, as funções consideradas são sempre não-negativas.
  - Uma função f é **assintoticamente não-negativa** se existe M tal que $f(n) \geq 0$ para todo n maior que M.
- Dependendo do caso, n representa diferentes valores:
  - Problemas de precisão arbitrária: número de bits.
  - Problemas em grafos: número de vértices e/ou arestas
  - Problemas com vetores: tamanho do vetor
  - Problemas de busca em textos: tamanho das strings

Notação assintótica

- Expressamos complexidade como uma função em n.
- A função mede o número de instruções de um algoritmo.
- Então, as funções consideradas são sempre não-negativas.
  - Uma função f é **assintoticamente não-negativa** se existe M tal que $f(n) \geq 0$ para todo n maior que M.
- Dependendo do caso, n representa diferentes valores:
  - Problemas de precisão arbitrária: número de bits.
  - Problemas em grafos: número de vértices e/ou arestas
  - Problemas com vetores: tamanho do vetor
  - Problemas de busca em textos: tamanho das strings

Notação assintótica

- Expressamos complexidade como uma função em n.
- A função mede o número de instruções de um algoritmo.
- Então, as funções consideradas são sempre não-negativas.
  - Uma função f é **assintoticamente não-negativa** se existe M tal que $f(n) \geq 0$ para todo n maior que M.
- Dependendo do caso, n representa diferentes valores:
  - Problemas de precisão arbitrária: número de bits.
  - Problemas em grafos: número de vértices e/ou arestas
  - Problemas com vetores: tamanho do vetor
  - Problemas de busca em textos: tamanho das strings

Notação assintótica

- Expressamos complexidade como uma função em n.
- A função mede o número de instruções de um algoritmo.
- Então, as funções consideradas são sempre não-negativas.
  - Uma função f é **assintoticamente não-negativa** se existe M tal que $f(n) \geq 0$ para todo n maior que M.
- Dependendo do caso, n representa diferentes valores:
  - Problemas de precisão arbitrária: número de bits.
  - Problemas em grafos: número de vértices e/ou arestas
  - Problemas com vetores: tamanho do vetor
  - Problemas de busca em textos: tamanho das strings

Notação assintótica

- Expressamos complexidade como uma função em n.
- A função mede o número de instruções de um algoritmo.
- Então, as funções consideradas são sempre não-negativas.
  - Uma função f é **assintoticamente não-negativa** se existe M tal que $f(n) \geq 0$ para todo n maior que M.
- Dependendo do caso, n representa diferentes valores:
  - Problemas de precisão arbitrária: número de bits.
  - Problemas em grafos: número de vértices e/ou arestas
  - Problemas com vetores: tamanho do vetor
  - Problemas de busca em textos: tamanho das strings

Notação assintótica

- Expressamos complexidade como uma função em n.
- A função mede o número de instruções de um algoritmo.
- Então, as funções consideradas são sempre não-negativas.
  - Uma função f é **assintoticamente não-negativa** se existe M tal que $f(n) \geq 0$ para todo n maior que M.
- Dependendo do caso, n representa diferentes valores:
  - Problemas de precisão arbitrária: número de bits.
  - Problemas em grafos: número de vértices e/ou arestas
  - Problemas com vetores: tamanho do vetor
  - Problemas de busca em textos: tamanho das strings

Notação assintótica

- Expressamos complexidade como uma função em n.
- A função mede o número de instruções de um algoritmo.
- Então, as funções consideradas são sempre não-negativas.
  - Uma função f é **assintoticamente não-negativa** se existe M tal que $f(n) \geq 0$ para todo n maior que M.
- Dependendo do caso, n representa diferentes valores:
  - Problemas de precisão arbitrária: número de bits.
  - Problemas em grafos: número de vértices e/ou arestas
  - Problemas com vetores: tamanho do vetor
  - Problemas de busca em textos: tamanho das strings

Complexidade assintótica

Estamos interessados principalmente

- na análise de pior caso
- no tempo de execução para instâncias grandes

Comportamento assintótico

- no pior caso temos $T(n) = an^2 + bn + c$
  - o termo dominante é o que contém n^2
  - o tempo de execução é uma função quadrática
  - as constantes a, b, c só dependem da implementação
- não nos preocupamos com os valores de a, b, c

Por que isso é razoável?

Complexidade assintótica

Estamos interessados principalmente

- na análise de pior caso
- no tempo de execução para instâncias grandes

Comportamento assintótico

- no pior caso temos $T(n) = an^2 + bn + c$
  - o termo dominante é o que contém n^2
  - o tempo de execução é uma função quadrática
  - as constantes a, b, c só dependem da implementação
- não nos preocupamos com os valores de a, b, c

Por que isso é razoável?

Complexidade assintótica

Estamos interessados principalmente

- na análise de pior caso
- no tempo de execução para instâncias grandes

Comportamento assintótico

- no pior caso temos $T(n) = an^2 + bn + c$
  - o termo dominante é o que contém n^2
  - o tempo de execução é uma função quadrática
  - as constantes a, b, c só dependem da implementação
- não nos preocupamos com os valores de a, b, c

Por que isso é razoável?

Complexidade assintótica

Estamos interessados principalmente

- na análise de pior caso
- no tempo de execução para instâncias grandes

Comportamento assintótico

- no pior caso temos $T(n) = an^2 + bn + c$
  - o termo dominante é o que contém n^2
  - o tempo de execução é uma função quadrática
  - as constantes a, b, c só dependem da implementação
- não nos preocupamos com os valores de a, b, c

Por que isso é razoável?

Complexidade assintótica

Estamos interessados principalmente

- na análise de pior caso
- no tempo de execução para instâncias grandes

Comportamento assintótico

- no pior caso temos $T(n) = an^2 + bn + c$
  - o termo dominante é o que contém n^2
  - o tempo de execução é uma função quadrática
  - as constantes a, b, c só dependem da implementação
- não nos preocupamos com os valores de a, b, c

Por que isso é razoável?

Complexidade assintótica

Estamos interessados principalmente

- na análise de pior caso
- no tempo de execução para instâncias grandes

Comportamento assintótico

- no pior caso temos $T(n) = an^2 + bn + c$
  - o termo dominante é o que contém n^2
  - o tempo de execução é uma função quadrática
  - as constantes a, b, c só dependem da implementação
- não nos preocupamos com os valores de a, b, c

Por que isso é razoável?

Complexidade assintótica

Estamos interessados principalmente

- na análise de pior caso
- no tempo de execução para instâncias grandes

Comportamento assintótico

- no pior caso temos $T(n) = an^2 + bn + c$
  - o termo dominante é o que contém n^2
  - o tempo de execução é uma função quadrática
  - as constantes a, b, c só dependem da implementação
- não nos preocupamos com os valores de a, b, c

Por que isso é razoável?

Complexidade assintótica

Estamos interessados principalmente

- na análise de pior caso
- no tempo de execução para instâncias grandes

Comportamento assintótico

- no pior caso temos $T(n) = an^2 + bn + c$
  - o termo dominante é o que contém n^2
  - o tempo de execução é uma função quadrática
  - as constantes a, b, c só dependem da implementação
- não nos preocupamos com os valores de a, b, c

Por que isso é razoável?

Complexidade assintótica

Estamos interessados principalmente

- na análise de pior caso
- no tempo de execução para instâncias grandes

Comportamento assintótico

- no pior caso temos $T(n) = an^2 + bn + c$
  - o termo dominante é o que contém n^2
  - o tempo de execução é uma função quadrática
  - as constantes a, b, c só dependem da implementação
- não nos preocupamos com os valores de a, b, c

Por que isso é razoável?

Complexidade assintótica

Estamos interessados principalmente

- na análise de pior caso
- no tempo de execução para instâncias grandes

Comportamento assintótico

- no pior caso temos $T(n) = an^2 + bn + c$
  - o termo dominante é o que contém n^2
  - o tempo de execução é uma função quadrática
  - as constantes a, b, c só dependem da implementação
- não nos preocupamos com os valores de a, b, c

Por que isso é razoável?

Um exemplo de função quadrática

Considere a função $3n^2 + 10n + 50$

n	$3n^2 + 10n + 50$	$3n^2$
64	12978	12288
128	50482	49152
512	791602	786432
1024	3156018	3145728
2048	12603442	12582912
4096	50372658	50331648
8192	201408562	201326592
16384	805470258	805306368
32768	3221553202	3221225472

- quando n é grande, o termo $3n^2$ é uma boa estimativa
- podemos nos concentrar nos termos dominantes

Um exemplo de função quadrática

Considere a função $3n^2 + 10n + 50$

n	$3n^2 + 10n + 50$	$3n^2$
64	12978	12288
128	50482	49152
512	791602	786432
1024	3156018	3145728
2048	12603442	12582912
4096	50372658	50331648
8192	201408562	201326592
16384	805470258	805306368
32768	3221553202	3221225472

- quando n é grande, o termo $3n^2$ é uma boa estimativa
- podemos nos concentrar nos termos dominantes

Um exemplo de função quadrática

Considere a função $3n^2 + 10n + 50$

n	$3n^2 + 10n + 50$	$3n^2$
64	12978	12288
128	50482	49152
512	791602	786432
1024	3156018	3145728
2048	12603442	12582912
4096	50372658	50331648
8192	201408562	201326592
16384	805470258	805306368
32768	3221553202	3221225472

- quando n é grande, o termo $3n^2$ é uma boa estimativa
- podemos nos concentrar nos termos dominantes

Um exemplo de função quadrática

Considere a função $3n^2 + 10n + 50$

n	$3n^2 + 10n + 50$	$3n^2$
64	12978	12288
128	50482	49152
512	791602	786432
1024	3156018	3145728
2048	12603442	12582912
4096	50372658	50331648
8192	201408562	201326592
16384	805470258	805306368
32768	3221553202	3221225472

- quando n é grande, o termo $3n^2$ é uma boa estimativa
- podemos nos concentrar nos termos dominantes

Notação assintótica e crescimento de funções

- Definições

Notação O (Big O)

- Considere a função $7n^3 + 100n^2 - 20n + 6$. Seu termo de ordem mais alta é $7n^3$ e, portanto, gostaríamos de dizer que a taxa de crescimento dessa função é n^3.
- Como essa função não cresce mais rápido que n^3, escrevemos que ela é $O(n^3)$.
- A notação O caracteriza um **limitante superior** no comportamento assintótico de uma função.

Notação O (Big O)

- Considere a função $7n^3 + 100n^2 - 20n + 6$. Seu termo de ordem mais alta é $7n^3$ e, portanto, gostaríamos de dizer que a taxa de crescimento dessa função é n^3.

- Como essa função não cresce mais rápido que n^3, escrevemos que ela é $O(n^3)$.

- A notação O caracteriza um **limitante superior** no comportamento assintótico de uma função.

Notação O (Big O)

- Considere a função $7n^3 + 100n^2 - 20n + 6$. Seu termo de ordem mais alta é $7n^3$ e, portanto, gostaríamos de dizer que a taxa de crescimento dessa função é n^3.
- Como essa função não cresce mais rápido que n^3, escrevemos que ela é $O(n^3)$.
- A notação O caracteriza um **limitante superior** no comportamento assintótico de uma função.

Notação O (Big O)

Definição

Para uma dada função $g(n)$, a classe $O(g(n))$ é o conjunto de funções $f(n)$ tais que:

- existem constantes positivas c e n_0 de modo que

$$0 \leq f(n) \leq c \cdot g(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce no máximo tão rápido quanto $g(n)$.

Notação O (Big O)

Definição

Para uma dada função $g(n)$, a classe $O(g(n))$ é o conjunto de funções $f(n)$ tais que:

- existem constantes positivas c e n_0 de modo que

$$0 \leq f(n) \leq c \cdot g(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce no máximo tão rápido quanto $g(n)$.

Notação O (Big O)

Definição

Para uma dada função $g(n)$, a classe $O(g(n))$ é o conjunto de funções $f(n)$ tais que:

- existem constantes positivas c e n_0 de modo que

$$0 \leq f(n) \leq c \cdot g(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce no máximo tão rápido quanto $g(n)$.

Notação O (Big O)

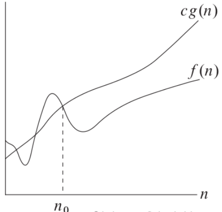
Definição

Para uma dada função $g(n)$, a classe $O(g(n))$ é o conjunto de funções $f(n)$ tais que:

- existem constantes positivas c e n_0 de modo que

$$0 \leq f(n) \leq c \cdot g(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce no máximo tão rápido quanto $g(n)$.



$f(n) = O(g(n))$

Notação O (Big O)

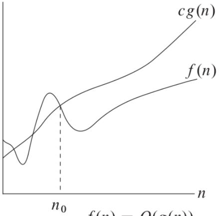
Definição

Para uma dada função $g(n)$, a classe $O(g(n))$ é o conjunto de funções $f(n)$ tais que:

- existem constantes positivas c e n_0 de modo que

$$0 \leq f(n) \leq c \cdot g(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce no máximo tão rápido quanto $g(n)$.



$f(n) = O(g(n))$

Geralmente, escrevemos $f(n) = O(g(n))$ para dizer que $f(n) \in O(g(n))$, ou dizemos apenas que $f(n)$ é $O(g(n))$.

Notação O: exemplo

Exemplo

Provar que $\frac{1}{2}n^2 - 3n = O(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$
\begin{aligned}
f(n) &= \frac{1}{2}n^2 - 3n \\
&= \frac{1}{2} \cdot \left(n^2 - 6n\right) \\
&\leq \frac{1}{2} \cdot n^2 && \text{(para } n \geq 0\text{)} \\
&= c \cdot g(n).
\end{aligned}
$$

- Logo, $f(n) \leq c \cdot g(n)$ para $c = \frac{1}{2}$ e $n \geq 0$.

Notação O: exemplo

Exemplo

Provar que $\frac{1}{2}n^2 - 3n = O(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$
\begin{aligned}
f(n) &= \frac{1}{2}n^2 - 3n \\
&= \frac{1}{2} \cdot \left(n^2 - 6n\right) \\
&\leq \frac{1}{2} \cdot n^2 && \text{(para } n \geq 0\text{)} \\
&= c \cdot g(n).
\end{aligned}
$$

- Logo, $f(n) \leq c \cdot g(n)$ para $c = \frac{1}{2}$ e $n \geq 0$.

Notação O: exemplo

Exemplo

Provar que $\frac{1}{2}n^2 - 3n = O(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$\begin{aligned} f(n) &= \frac{1}{2}n^2 - 3n \\ &= \frac{1}{2} \cdot \left(n^2 - 6n\right) \\ &\le \frac{1}{2} \cdot n^2 && \text{(para } n \ge 0\text{)} \\ &= c \cdot g(n). \end{aligned}$$

- Logo, $f(n) \le c \cdot g(n)$ para $c = \frac{1}{2}$ e $n \ge 0$.

Notação O: exemplo

Exemplo

Provar que $\frac{1}{2}n^2 - 3n = O(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$\begin{aligned} f(n) &= \frac{1}{2}n^2 - 3n \\ &= \frac{1}{2} \cdot \left(n^2 - 6n\right) \\ &\leq \frac{1}{2} \cdot n^2 && \text{(para } n \geq 0\text{)} \\ &= c \cdot g(n). \end{aligned}$$

- Logo, $f(n) \leq c \cdot g(n)$ para $c = \frac{1}{2}$ e $n \geq 0$.

Notação O: exemplo

Exemplo

Provar que $\frac{1}{2}n^2 - 3n = O(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$\begin{aligned} f(n) &= \frac{1}{2}n^2 - 3n \\ &= \frac{1}{2} \cdot \left(n^2 - 6n\right) \\ &\leq \frac{1}{2} \cdot n^2 && \text{(para } n \geq 0) \\ &= c \cdot g(n). \end{aligned}$$

- Logo, $f(n) \leq c \cdot g(n)$ para $c = \frac{1}{2}$ e $n \geq 0$.

Notação O: exemplo

Exemplo

Provar que $\frac{1}{2}n^2 - 3n = O(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$
\begin{aligned}
f(n) &= \frac{1}{2}n^2 - 3n \\
&= \frac{1}{2} \cdot \left(n^2 - 6n\right) \\
&\leq \frac{1}{2} \cdot n^2 && \text{(para } n \geq 0\text{)} \\
&= c \cdot g(n).
\end{aligned}
$$

- Logo, $f(n) \leq c \cdot g(n)$ para $c = \frac{1}{2}$ e $n \geq 0$.

Notação O: exemplo

Exemplo

Provar que $\frac{1}{2}n^2 - 3n = O(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$\begin{aligned} f(n) &= \frac{1}{2}n^2 - 3n \\ &= \frac{1}{2} \cdot \left(n^2 - 6n\right) \\ &\leq \frac{1}{2} \cdot n^2 && (\text{para } n \geq 0) \\ &= c \cdot g(n). \end{aligned}$$

- Logo, $f(n) \leq c \cdot g(n)$ para $c = \frac{1}{2}$ e $n \geq 0$.

Notação O: exemplo

Exemplo

Provar que $\frac{1}{2}n^2 - 3n = O(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$
\begin{aligned}
f(n) &= \frac{1}{2}n^2 - 3n \\
&= \frac{1}{2} \cdot \left(n^2 - 6n\right) \\
&\le \frac{1}{2} \cdot n^2 && (\text{para } n \ge 0) \\
&= c \cdot g(n).
\end{aligned}
$$

- Logo, $f(n) \le c \cdot g(n)$ para $c = \frac{1}{2}$ e $n \ge 0$.

Notação O: exemplo

Exemplo

Provar que $\frac{1}{2}n^2 - 3n = O(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$\begin{aligned} f(n) &= \frac{1}{2}n^2 - 3n \\ &= \frac{1}{2} \cdot \left(n^2 - 6n\right) \\ &\leq \frac{1}{2} \cdot n^2 & \text{(para } n \geq 0) \\ &= c \cdot g(n). \end{aligned}$$

- Logo, $f(n) \leq c \cdot g(n)$ para $c = \frac{1}{2}$ e $n \geq 0$.

Notação O: exemplo

Exemplo

Provar que $4n^2 + 100n + 500 = O(n^2)$

- Temos $f(n) = 4n^2 + 100n + 500$ e $g(n) = n^2$.
- Precisamos encontrar constantes c e n_0 tais que $4n^2 + 100n + 500 \leq c \cdot n^2$ para todo $n \geq n_0$.
- Note que $100n \leq 100n^2$ para todo $n \geq 0$ e que $500 \leq 500n^2$ para todo $n \geq 1$.
- Logo, $4n^2 + 100n + 500 \leq 4n^2 + 100n^2 + 500n^2 = 604n^2$ para todo $n \geq 1$
- Assim, $f(n) \leq c \cdot g(n)$ para $c = 604$ e $n \geq n_0 = 1$

Notação O: exemplo

Exemplo

Provar que $4n^2 + 100n + 500 = O(n^2)$

- Temos $f(n) = 4n^2 + 100n + 500$ e $g(n) = n^2$.
- Precisamos encontrar constantes c e n_0 tais que $4n^2 + 100n + 500 \leq c \cdot n^2$ para todo $n \geq n_0$.
- Note que $100n \leq 100n^2$ para todo $n \geq 0$ e que $500 \leq 500n^2$ para todo $n \geq 1$.
- Logo, $4n^2 + 100n + 500 \leq 4n^2 + 100n^2 + 500n^2 = 604n^2$ para todo $n \geq 1$
- Assim, $f(n) \leq c \cdot g(n)$ para $c = 604$ e $n \geq n_0 = 1$

Notação O: exemplo

Exemplo

Provar que $4n^2 + 100n + 500 = O(n^2)$

- Temos $f(n) = 4n^2 + 100n + 500$ e $g(n) = n^2$.
- Precisamos encontrar constantes c e n_0 tais que $4n^2 + 100n + 500 \leq c \cdot n^2$ para todo $n \geq n_0$.
- Note que $100n \leq 100n^2$ para todo $n \geq 0$ e que $500 \leq 500n^2$ para todo $n \geq 1$.
- Logo, $4n^2 + 100n + 500 \leq 4n^2 + 100n^2 + 500n^2 = 604n^2$ para todo $n \geq 1$
- Assim, $f(n) \leq c \cdot g(n)$ para $c = 604$ e $n \geq n_0 = 1$

Notação O: exemplo

Exemplo

Provar que $4n^2 + 100n + 500 = O(n^2)$

- Temos $f(n) = 4n^2 + 100n + 500$ e $g(n) = n^2$.
- Precisamos encontrar constantes c e n_0 tais que $4n^2 + 100n + 500 \leq c \cdot n^2$ para todo $n \geq n_0$.
- Note que $100n \leq 100n^2$ para todo $n \geq 0$ e que $500 \leq 500n^2$ para todo $n \geq 1$.
- Logo, $4n^2 + 100n + 500 \leq 4n^2 + 100n^2 + 500n^2 = 604n^2$ para todo $n \geq 1$
- Assim, $f(n) \leq c \cdot g(n)$ para $c = 604$ e $n \geq n_0 = 1$

Notação O: exemplo

Exemplo

Provar que $4n^2 + 100n + 500 = O(n^2)$

- Temos $f(n) = 4n^2 + 100n + 500$ e $g(n) = n^2$.
- Precisamos encontrar constantes c e n_0 tais que $4n^2 + 100n + 500 \leq c \cdot n^2$ para todo $n \geq n_0$.
- Note que $100n \leq 100n^2$ para todo $n \geq 0$ e que $500 \leq 500n^2$ para todo $n \geq 1$.
- Logo, $4n^2 + 100n + 500 \leq 4n^2 + 100n^2 + 500n^2 = 604n^2$ para todo $n \geq 1$
- Assim, $f(n) \leq c \cdot g(n)$ para $c = 604$ e $n \geq n_0 = 1$

Notação O: exemplo

Exemplo

Provar que $4n^2 + 100n + 500 = O(n^2)$

- Temos $f(n) = 4n^2 + 100n + 500$ e $g(n) = n^2$.
- Precisamos encontrar constantes c e n_0 tais que $4n^2 + 100n + 500 \leq c \cdot n^2$ para todo $n \geq n_0$.
- Note que $100n \leq 100n^2$ para todo $n \geq 0$ e que $500 \leq 500n^2$ para todo $n \geq 1$.
- Logo, $4n^2 + 100n + 500 \leq 4n^2 + 100n^2 + 500n^2 = 604n^2$ para todo $n \geq 1$
- Assim, $f(n) \leq c \cdot g(n)$ para $c = 604$ e $n \geq n_0 = 1$

Notação O: exemplo

Exemplo

Provar que $3n^3 + 20n^2 + 5 = O(n^3)$

- É preciso encontrar c e n_0 positivos tais que $3n^3 + 20n^2 + 5 \leq cn^3$ para todo $n \geq n_0$.
- Como $3n^3 + 20n^2 + 5 \leq (3 + 20 + 5)n^3$ para todo $n \geq 1$, podemos tomar $c = 28$ e qualquer $n_0 \geq 1$.

Notação O: exemplo

Exemplo

Provar que $3n^3 + 20n^2 + 5 = O(n^3)$

- É preciso encontrar c e n_0 positivos tais que $3n^3 + 20n^2 + 5 \leq cn^3$ para todo $n \geq n_0$.
- Como $3n^3 + 20n^2 + 5 \leq (3 + 20 + 5)n^3$ para todo $n \geq 1$, podemos tomar $c = 28$ e qualquer $n_0 \geq 1$.

Notação O: exemplo

Exemplo

Provar que $3n^3 + 20n^2 + 5 = O(n^3)$

- É preciso encontrar c e n_0 positivos tais que $3n^3 + 20n^2 + 5 \leq cn^3$ para todo $n \geq n_0$.
- Como $3n^3 + 20n^2 + 5 \leq (3 + 20 + 5)n^3$ para todo $n \geq 1$, podemos tomar $c = 28$ e qualquer $n_0 \geq 1$.

Notação O: exemplo

Exemplo

Provar que $3 \lg n + 5 = O(\lg n)$

- É preciso encontrar c e n_0 positivos tais que $3 \lg n + 5 \leq c \lg n$ para todo $n \geq n_0$.
- Note que $3 \lg n + 5 \leq (3+5) \lg n$ se $n > 1$ ($\lg 1 = 0$).
- basta tomar, por exemplo, $c = 8$ e $n_0 = 2$.

Notação O: exemplo

Exemplo

Provar que $3 \lg n + 5 = O(\lg n)$

- É preciso encontrar c e n_0 positivos tais que $3 \lg n + 5 \leq c \lg n$ para todo $n \geq n_0$.
- Note que $3 \lg n + 5 \leq (3+5) \lg n$ se $n > 1$ ($\lg 1 = 0$).
- basta tomar, por exemplo, $c = 8$ e $n_0 = 2$.

Notação O: exemplo

Exemplo

Provar que $3 \lg n + 5 = O(\lg n)$

- É preciso encontrar c e n_0 positivos tais que $3 \lg n + 5 \leq c \lg n$ para todo $n \geq n_0$.
- Note que $3 \lg n + 5 \leq (3+5) \lg n$ se $n > 1$ ($\lg 1 = 0$).
- basta tomar, por exemplo, $c = 8$ e $n_0 = 2$.

Notação O: exemplo

Exemplo

Provar que $3 \lg n + 5 = O(\lg n)$

- É preciso encontrar c e n_0 positivos tais que $3 \lg n + 5 \leq c \lg n$ para todo $n \geq n_0$.
- Note que $3 \lg n + 5 \leq (3 + 5) \lg n$ se $n > 1$ ($\lg 1 = 0$).
- basta tomar, por exemplo, $c = 8$ e $n_0 = 2$.

Notação O: exemplo

Exemplo

Provar que $n^3 - 100n^2 \neq O(n^2)$

- Suponha, por absurdo, que $n^3 - 100n^2 = O(n^2)$.
- Então, existem constantes positivas c e n_0 tais que $n^3 - 100n^2 \leq cn^2$ para todo $n \geq n_0$.
- Dividimos ambos os lados por n^2, obtendo $n - 100 \leq c$.
- Independentemente do valor que escolhermos para a constante c, esta desigualdade não vale para qualquer valor de $n > c + 100$. Isso contradiz a suposição de que $n^3 - 100n^2 = O(n^2)$.
- Portanto, $n^3 - 100n^2 \neq O(n^2)$.

Notação O: exemplo

Exemplo

Provar que $n^3 - 100n^2 \neq O(n^2)$

- Suponha, por absurdo, que $n^3 - 100n^2 = O(n^2)$.
- Então, existem constantes positivas c e n_0 tais que $n^3 - 100n^2 \leq cn^2$ para todo $n \geq n_0$.
- Dividimos ambos os lados por n^2, obtendo $n - 100 \leq c$.
- Independentemente do valor que escolhermos para a constante c, esta desigualdade não vale para qualquer valor de $n > c + 100$. Isso contradiz a suposição de que $n^3 - 100n^2 = O(n^2)$.
- Portanto, $n^3 - 100n^2 \neq O(n^2)$.

Notação O: exemplo

Exemplo

Provar que $n^3 - 100n^2 \neq O(n^2)$

- Suponha, por absurdo, que $n^3 - 100n^2 = O(n^2)$.
- Então, existem constantes positivas c e n_0 tais que $n^3 - 100n^2 \leq cn^2$ para todo $n \geq n_0$.
- Dividimos ambos os lados por n^2, obtendo $n - 100 \leq c$.
- Independentemente do valor que escolhermos para a constante c, esta desigualdade não vale para qualquer valor de $n > c + 100$. Isso contradiz a suposição de que $n^3 - 100n^2 = O(n^2)$.
- Portanto, $n^3 - 100n^2 \neq O(n^2)$.

Notação O: exemplo

Exemplo

Provar que $n^3 - 100n^2 \neq O(n^2)$

- Suponha, por absurdo, que $n^3 - 100n^2 = O(n^2)$.
- Então, existem constantes positivas c e n_0 tais que $n^3 - 100n^2 \leq cn^2$ para todo $n \geq n_0$.
- Dividimos ambos os lados por n^2, obtendo $n - 100 \leq c$.
- Independentemente do valor que escolhermos para a constante c, esta desigualdade não vale para qualquer valor de $n > c + 100$. Isso contradiz a suposição de que $n^3 - 100n^2 = O(n^2)$.
- Portanto, $n^3 - 100n^2 \neq O(n^2)$.

Notação O: exemplo

Exemplo

Provar que $n^3 - 100n^2 \neq O(n^2)$

- Suponha, por absurdo, que $n^3 - 100n^2 = O(n^2)$.
- Então, existem constantes positivas c e n_0 tais que $n^3 - 100n^2 \leq cn^2$ para todo $n \geq n_0$.
- Dividimos ambos os lados por n^2, obtendo $n - 100 \leq c$.
- Independentemente do valor que escolhermos para a constante c, esta desigualdade não vale para qualquer valor de $n > c + 100$. Isso contradiz a suposição de que $n^3 - 100n^2 = O(n^2)$.
- Portanto, $n^3 - 100n^2 \neq O(n^2)$.

Notação O: exemplo

Exemplo

Provar que $n^3 - 100n^2 \neq O(n^2)$

- Suponha, por absurdo, que $n^3 - 100n^2 = O(n^2)$.
- Então, existem constantes positivas c e n_0 tais que $n^3 - 100n^2 \leq cn^2$ para todo $n \geq n_0$.
- Dividimos ambos os lados por n^2, obtendo $n - 100 \leq c$.
- Independentemente do valor que escolhermos para a constante c, esta desigualdade não vale para qualquer valor de $n > c + 100$. Isso contradiz a suposição de que $n^3 - 100n^2 = O(n^2)$.
- Portanto, $n^3 - 100n^2 \neq O(n^2)$.

Notação O: exemplo

Exemplo

Provar que $\lceil \lg n \rceil = O(n)$

- Tomando valores pequenos de n vemos que:
  - $n = 1:$ $\lceil \lg n \rceil = \lceil \lg 1 \rceil = 0 \leq 1 = n$
  - $n = 2:$ $\lceil \lg n \rceil = \lceil \lg 2 \rceil = 1 \leq 2 = n$
  - $n = 3:$ $\lceil \lg n \rceil = \lceil \lg 3 \rceil = 2 \leq 3 = n$
  - $n = 4:$ $\lceil \lg n \rceil = \lceil \lg 4 \rceil = 2 \leq 4 = n$
- **Conjectura:** $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$.
- Se provarmos essa conjectura, teremos provado que $\lceil \lg n \rceil = O(n)$. Por quê??
- Vamos fazer uma prova por indução. Força!

Notação O: exemplo

Exemplo

Provar que $\lceil \lg n \rceil = O(n)$

- Tomando valores pequenos de n vemos que:
  - $n = 1:$ $\lceil \lg n \rceil = \lceil \lg 1 \rceil = 0 \leq 1 = n$
  - $n = 2:$ $\lceil \lg n \rceil = \lceil \lg 2 \rceil = 1 \leq 2 = n$
  - $n = 3:$ $\lceil \lg n \rceil = \lceil \lg 3 \rceil = 2 \leq 3 = n$
  - $n = 4:$ $\lceil \lg n \rceil = \lceil \lg 4 \rceil = 2 \leq 4 = n$
- **Conjectura:** $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$.
- Se provarmos essa conjectura, teremos provado que $\lceil \lg n \rceil = O(n)$. Por quê??
- Vamos fazer uma prova por indução. Força!

Notação O: exemplo

Exemplo

Provar que $\lceil \lg n \rceil = O(n)$

- Tomando valores pequenos de n vemos que:
  - $n = 1:$ $\quad \lceil \lg n \rceil = \lceil \lg 1 \rceil = 0 \leq 1 = n$
  - $n = 2:$ $\quad \lceil \lg n \rceil = \lceil \lg 2 \rceil = 1 \leq 2 = n$
  - $n = 3:$ $\quad \lceil \lg n \rceil = \lceil \lg 3 \rceil = 2 \leq 3 = n$
  - $n = 4:$ $\quad \lceil \lg n \rceil = \lceil \lg 4 \rceil = 2 \leq 4 = n$
- **Conjectura:** $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$.
- Se provarmos essa conjectura, teremos provado que $\lceil \lg n \rceil = O(n)$. Por quê??
- Vamos fazer uma prova por indução. Força!

Notação O: exemplo

Exemplo

Provar que $\lceil \lg n \rceil = O(n)$

- Tomando valores pequenos de n vemos que:
  - $n = 1:$ $\quad \lceil \lg n \rceil = \lceil \lg 1 \rceil = 0 \leq 1 = n$
  - $n = 2:$ $\quad \lceil \lg n \rceil = \lceil \lg 2 \rceil = 1 \leq 2 = n$
  - $n = 3:$ $\quad \lceil \lg n \rceil = \lceil \lg 3 \rceil = 2 \leq 3 = n$
  - $n = 4:$ $\quad \lceil \lg n \rceil = \lceil \lg 4 \rceil = 2 \leq 4 = n$
- **Conjectura:** $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$.
- Se provarmos essa conjectura, teremos provado que $\lceil \lg n \rceil = O(n)$. Por quê??
- Vamos fazer uma prova por indução. Força!

Notação O: exemplo

Exemplo

Provar que $\lceil \lg n \rceil = O(n)$

- Tomando valores pequenos de n vemos que:
  - $n = 1 :\qquad \lceil \lg n \rceil = \lceil \lg 1 \rceil = 0 \leq 1 = n$
  - $n = 2 :\qquad \lceil \lg n \rceil = \lceil \lg 2 \rceil = 1 \leq 2 = n$
  - $n = 3 :\qquad \lceil \lg n \rceil = \lceil \lg 3 \rceil = 2 \leq 3 = n$
  - $n = 4 :\qquad \lceil \lg n \rceil = \lceil \lg 4 \rceil = 2 \leq 4 = n$
- **Conjectura:** $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$.
- Se provarmos essa conjectura, teremos provado que $\lceil \lg n \rceil = O(n)$. Por quê??
- Vamos fazer uma prova por indução. Força!

Notação O: exemplo

Exemplo

Provar que $\lceil \lg n \rceil = O(n)$

- Tomando valores pequenos de n vemos que:
  - $n = 1 :$ $\lceil \lg n \rceil = \lceil \lg 1 \rceil = 0 \leq 1 = n$
  - $n = 2 :$ $\lceil \lg n \rceil = \lceil \lg 2 \rceil = 1 \leq 2 = n$
  - $n = 3 :$ $\lceil \lg n \rceil = \lceil \lg 3 \rceil = 2 \leq 3 = n$
  - $n = 4 :$ $\lceil \lg n \rceil = \lceil \lg 4 \rceil = 2 \leq 4 = n$
- **Conjectura:** $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$.
- Se provarmos essa conjectura, teremos provado que $\lceil \lg n \rceil = O(n)$. Por quê??
- Vamos fazer uma prova por indução. Força!

Notação O: exemplo

Exemplo

Provar que $\lceil \lg n \rceil = O(n)$

- Tomando valores pequenos de n vemos que:
  - $n = 1: \quad \lceil \lg n \rceil = \lceil \lg 1 \rceil = 0 \leq 1 = n$
  - $n = 2: \quad \lceil \lg n \rceil = \lceil \lg 2 \rceil = 1 \leq 2 = n$
  - $n = 3: \quad \lceil \lg n \rceil = \lceil \lg 3 \rceil = 2 \leq 3 = n$
  - $n = 4: \quad \lceil \lg n \rceil = \lceil \lg 4 \rceil = 2 \leq 4 = n$
- **Conjectura:** $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$.
- Se provarmos essa conjectura, teremos provado que $\lceil \lg n \rceil = O(n)$. Por quê??
- Vamos fazer uma prova por indução. Força!

Notação O: exemplo

Exemplo

Provar que $\lceil \lg n \rceil = O(n)$

- Tomando valores pequenos de n vemos que:
  - $n = 1: \quad \lceil \lg n \rceil = \lceil \lg 1 \rceil = 0 \leq 1 = n$
  - $n = 2: \quad \lceil \lg n \rceil = \lceil \lg 2 \rceil = 1 \leq 2 = n$
  - $n = 3: \quad \lceil \lg n \rceil = \lceil \lg 3 \rceil = 2 \leq 3 = n$
  - $n = 4: \quad \lceil \lg n \rceil = \lceil \lg 4 \rceil = 2 \leq 4 = n$
- **Conjectura:** $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$.
- Se provarmos essa conjectura, teremos provado que $\lceil \lg n \rceil = O(n)$. Por quê??
- Vamos fazer uma prova por indução. Força!

Notação O: exemplo

Exemplo

Provar que $\lceil \lg n \rceil = O(n)$

- Tomando valores pequenos de n vemos que:
  - $n = 1: \qquad \lceil \lg n \rceil = \lceil \lg 1 \rceil = 0 \leq 1 = n$
  - $n = 2: \qquad \lceil \lg n \rceil = \lceil \lg 2 \rceil = 1 \leq 2 = n$
  - $n = 3: \qquad \lceil \lg n \rceil = \lceil \lg 3 \rceil = 2 \leq 3 = n$
  - $n = 4: \qquad \lceil \lg n \rceil = \lceil \lg 4 \rceil = 2 \leq 4 = n$
- **Conjectura:** $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$.
- Se provarmos essa conjectura, teremos provado que $\lceil \lg n \rceil = O(n)$. Por quê??
- Vamos fazer uma prova por indução. Força!

Notação O: exemplo (cont.)

Exemplo

Provar que $\lceil \lg n \rceil \le n$ para todo $n \ge 1$

- Prova por indução em n.
- **Caso básico:** $n = 1$. Neste caso, $\lceil \lg 1 \rceil = 0 \le 1$.
- **Passo indutivo:**
  - Agora, suponha que $n > 1$ e que $\lceil \lg(n-1) \rceil \le n-1$.
  - Então, temos que

$$\begin{aligned} \lceil \lg n \rceil &\le \lceil \lg(n-1) \rceil + 1 \\ &\le (n-1) + 1 && \text{(por hipótese de indução)} \\ &= n. \end{aligned}$$

Portanto, $\lceil \lg n \rceil \le n$ para todo $n \ge 1$. ■

Notação O: exemplo (cont.)

Exemplo

Provar que $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$

- Prova por indução em n.
- **Caso básico:** $n = 1$. Neste caso, $\lceil \lg 1 \rceil = 0 \leq 1$.
- **Passo indutivo:**
  - Agora, suponha que $n > 1$ e que $\lceil \lg(n-1) \rceil \leq n-1$.
  - Então, temos que

$$\begin{aligned} \lceil \lg n \rceil &\leq \lceil \lg(n-1) \rceil + 1 \\ &\leq (n-1) + 1 \qquad \text{(por hipótese de indução)} \\ &= n. \end{aligned}$$

Portanto, $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$. ■

Notação O: exemplo (cont.)

Exemplo

Provar que $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$

- Prova por indução em n.
- **Caso básico:** $n = 1$. Neste caso, $\lceil \lg 1 \rceil = 0 \leq 1$.
- **Passo indutivo:**
  - Agora, suponha que $n > 1$ e que $\lceil \lg(n-1) \rceil \leq n-1$.
  - Então, temos que

$$
\begin{aligned}
\lceil \lg n \rceil &\leq \lceil \lg(n-1) \rceil + 1 \\
&\leq (n-1) + 1 && \text{(por hipótese de indução)} \\
&= n.
\end{aligned}
$$

Portanto, $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$. ■

Notação O: exemplo (cont.)

Exemplo

Provar que $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$

- Prova por indução em n.
- **Caso básico:** $n = 1$. Neste caso, $\lceil \lg 1 \rceil = 0 \leq 1$.
- **Passo indutivo:**
  - Agora, suponha que $n > 1$ e que $\lceil \lg(n-1) \rceil \leq n - 1$.
  - Então, temos que

$$\begin{aligned} \lceil \lg n \rceil &\leq \lceil \lg(n-1) \rceil + 1 \\ &\leq (n-1) + 1 && \text{(por hipótese de indução)} \\ &= n. \end{aligned}$$

Portanto, $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$. ■

Notação O: exemplo (cont.)

Exemplo

Provar que $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$

- Prova por indução em n.
- **Caso básico:** $n = 1$. Neste caso, $\lceil \lg 1 \rceil = 0 \leq 1$.
- **Passo indutivo:**
  - Agora, suponha que $n > 1$ e que $\lceil \lg(n-1) \rceil \leq n-1$.
  - Então, temos que

$$
\begin{aligned}
\lceil \lg n \rceil &\leq \lceil \lg(n-1) \rceil + 1 \\
&\leq (n-1) + 1 \qquad \text{(por hipótese de indução)} \\
&= n.
\end{aligned}
$$

Portanto, $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$. ■

Notação O: exemplo (cont.)

Exemplo

Provar que $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$

- Prova por indução em n.
- **Caso básico:** $n = 1$. Neste caso, $\lceil \lg 1 \rceil = 0 \leq 1$.
- **Passo indutivo:**
  - Agora, suponha que $n > 1$ e que $\lceil \lg(n-1) \rceil \leq n - 1$.
  - Então, temos que

$$\begin{aligned} \lceil \lg n \rceil &\leq \lceil \lg(n-1) \rceil + 1 \\ &\leq (n-1) + 1 \qquad \text{(por hipótese de indução)} \\ &= n. \end{aligned}$$

Portanto, $\lceil \lg n \rceil \leq n$ para todo $n \geq 1$. ■

Notação de igualdade para conjuntos de funções

- O sinal de igualdade é utilizado no sentido de "representatividade" e pode ser lido como "é".
- Um conjunto em uma fórmula representa uma função anônima.
- Exemplo: $f(n) = n^3 + O(n^2)$

  significa que existe um $h(n) \in O(n^2)$ de forma que $f(n) = n^3 + h(n)$.
- Exemplo: $n^2 + O(n) = O(n^2)$

  significa que, para qualquer $f(n) \in O(n)$ existe $h(n) \in O(n^2)$ de forma que $n^2 + f(n) = h(n)$.

Notação de igualdade para conjuntos de funções

- O sinal de igualdade é utilizado no sentido de “representatividade” e pode ser lido como “é”.
- Um conjunto em uma fórmula representa uma função anônima.
- Exemplo: $f(n) = n^3 + O(n^2)$

  significa que existe um $h(n) \in O(n^2)$ de forma que $f(n) = n^3 + h(n)$.
- Exemplo: $n^2 + O(n) = O(n^2)$

  significa que, para qualquer $f(n) \in O(n)$ existe $h(n) \in O(n^2)$ de forma que $n^2 + f(n) = h(n)$.

Notação de igualdade para conjuntos de funções

- O sinal de igualdade é utilizado no sentido de "representatividade" e pode ser lido como "é".
- Um conjunto em uma fórmula representa uma função anônima.
- Exemplo: $f(n) = n^3 + O(n^2)$

  significa que existe um $h(n) \in O(n^2)$ de forma que $f(n) = n^3 + h(n)$.
- Exemplo: $n^2 + O(n) = O(n^2)$

  significa que, para qualquer $f(n) \in O(n)$ existe $h(n) \in O(n^2)$ de forma que $n^2 + f(n) = h(n)$.

Notação de igualdade para conjuntos de funções

- O sinal de igualdade é utilizado no sentido de "representatividade" e pode ser lido como "é".
- Um conjunto em uma fórmula representa uma função anônima.
- Exemplo: $f(n) = n^3 + O(n^2)$

  significa que existe um $h(n) \in O(n^2)$ de forma que $f(n) = n^3 + h(n)$.
- Exemplo: $n^2 + O(n) = O(n^2)$

  significa que, para qualquer $f(n) \in O(n)$ existe $h(n) \in O(n^2)$ de forma que $n^2 + f(n) = h(n)$.

Propriedades da classe O

Teorema

Suponha que f e g sejam duas funções tais que, para alguma outra função h, temos que $f = O(h)$ e $g = O(h)$. Então, $f + g = O(h)$.

Corolário

Seja k uma constante fixa e sejam $f_1, f_2, \ldots, f_k$ e h funções tais que $f_i = O(h)$ para todo i. Então $f_1 + f_2 + \cdots + f_k = O(h)$.

Propriedades da classe O

Teorema

Suponha que f e g sejam duas funções tais que, para alguma outra função h, temos que $f = O(h)$ e $g = O(h)$. Então, $f + g = O(h)$.

Corolário

Seja k uma constante fixa e sejam $f_1, f_2, \ldots, f_k$ e h funções tais que $f_i = O(h)$ para todo i. Então $f_1 + f_2 + \cdots + f_k = O(h)$.

Propriedades da classe O

Ao analisarmos um algoritmo, às vezes é fácil mostrar que uma das duas partes é mais lenta do que o outra. Gostaríamos de poder dizer que o tempo de execução de todo o algoritmo é assintoticamente comparável ao tempo de execução da parte lenta. Como o tempo total de execução é uma soma de duas funções, resultados em limites assintóticos para somas de funções são diretamente relevantes.

Corolário

Sejam f e g funções assintoticamente positivas tais que $g = O(f)$. Então $f + g = O(f)$.

Propriedades da classe O

Teorema: Somas e Multiplicações

- Se $f(n) = O(s(n))$ e $g(n) = O(r(n))$, então $f(n) + g(n) = O(s(n) + r(n))$.
- Se $f(n) = O(s(n))$ e $g(n) = O(r(n))$, então $f(n) \cdot g(n) = O(s(n) \cdot r(n))$.

Exercício: provar essas afirmações.

Propriedades da classe O

Teorema: Somas e Multiplicações

- Se $f(n) = O(s(n))$ e $g(n) = O(r(n))$, então $f(n) + g(n) = O(s(n) + r(n))$.
- Se $f(n) = O(s(n))$ e $g(n) = O(r(n))$, então $f(n) \cdot g(n) = O(s(n) \cdot r(n))$.

Exercício: provar essas afirmações.

Propriedades da classe O

Teorema: Somas e Multiplicações

- Se $f(n) = O(s(n))$ e $g(n) = O(r(n))$, então $f(n) + g(n) = O(s(n) + r(n))$.
- Se $f(n) = O(s(n))$ e $g(n) = O(r(n))$, então $f(n) \cdot g(n) = O(s(n) \cdot r(n))$.

Exercício: provar essas afirmações.

Propriedades da classe O

Teorema: Somas e Multiplicações

- Se $f(n) = O(s(n))$ e $g(n) = O(r(n))$, então $f(n) + g(n) = O(s(n) + r(n))$.
- Se $f(n) = O(s(n))$ e $g(n) = O(r(n))$, então $f(n) \cdot g(n) = O(s(n) \cdot r(n))$.

Exercício: provar essas afirmações.

Notação Ω

Definição

Para uma dada função $g(n)$, a classe $\Omega(g(n))$ é o conjunto de funções $f(n)$ tais que:

- existem constantes positivas c e n_0 de modo que

$$0 \leq cg(n) \leq f(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce no mínimo tão rápido quanto $g(n)$.

Notação Ω

Definição

Para uma dada função $g(n)$, a classe $\Omega(g(n))$ é o conjunto de funções $f(n)$ tais que:

- existem constantes positivas c e n_0 de modo que

$$0 \leq cg(n) \leq f(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce no mínimo tão rápido quanto $g(n)$.

Notação Ω

Definição

Para uma dada função $g(n)$, a classe $\Omega(g(n))$ é o conjunto de funções $f(n)$ tais que:

- existem constantes positivas c e n_0 de modo que

$$0 \leq cg(n) \leq f(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce no mínimo tão rápido quanto $g(n)$.

Notação Ω

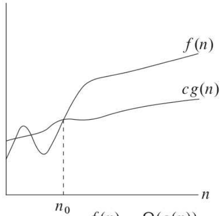
Definição

Para uma dada função $g(n)$, a classe $\Omega(g(n))$ é o conjunto de funções $f(n)$ tais que:

- existem constantes positivas c e n_0 de modo que

$$0 \leq cg(n) \leq f(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce no mínimo tão rápido quanto $g(n)$.



$f(n) = \Omega(g(n))$

Notação Ω

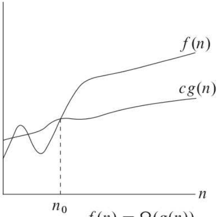
Definição

Para uma dada função $g(n)$, a classe $\Omega(g(n))$ é o conjunto de funções $f(n)$ tais que:

- existem constantes positivas c e n_0 de modo que

$$0 \leq cg(n) \leq f(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce no mínimo tão rápido quanto $g(n)$.



$f(n) = \Omega(g(n))$

Escrevemos $f(n) = \Omega(g(n))$ quando $f(n) \in \Omega(g(n))$.

Notação Ω: exemplo

Exemplo

$\frac{1}{2}n^2 - 3n \in \Omega(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$\begin{aligned}
f(n) &= \frac{1}{2}n^2 - 3n \\
&= \frac{n}{2}\cdot(n-6) \\
&\geq \frac{n}{2}\cdot\frac{n}{2} && \text{(para } n \geq 12\text{)} \\
&= c\cdot g(n). && \text{(para } c = \tfrac{1}{4}\text{)}
\end{aligned}$$

Notação Ω: exemplo

Exemplo

$\frac{1}{2}n^2 - 3n \in \Omega(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$
\begin{aligned}
f(n) &= \frac{1}{2}n^2 - 3n \\
&= \frac{n}{2} \cdot (n - 6) \\
&\geq \frac{n}{2} \cdot \frac{n}{2} && \text{(para } n \geq 12\text{)} \\
&= c \cdot g(n). && \text{(para } c = \frac{1}{4}\text{)}
\end{aligned}
$$

Notação Ω: exemplo

Exemplo

$\frac{1}{2}n^2 - 3n \in \Omega(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$
\begin{aligned}
f(n) &= \frac{1}{2}n^2 - 3n \\
&= \frac{n}{2} \cdot (n - 6) \\
&\geq \frac{n}{2} \cdot \frac{n}{2} && (\text{para } n \geq 12) \\
&= c \cdot g(n). && (\text{para } c = \frac{1}{4})
\end{aligned}
$$

Notação Ω: exemplo

Exemplo

$\frac{1}{2}n^2 - 3n \in \Omega(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$
\begin{aligned}
f(n) &= \frac{1}{2}n^2 - 3n \\
&= \frac{n}{2} \cdot (n - 6) \\
&\geq \frac{n}{2} \cdot \frac{n}{2} && (\text{para } n \geq 12) \\
&= c \cdot g(n). && (\text{para } c = \frac{1}{4})
\end{aligned}
$$

Notação Ω: exemplo

Exemplo

$\frac{1}{2}n^2 - 3n \in \Omega(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$
\begin{aligned}
f(n) &= \frac{1}{2}n^2 - 3n \\
&= \frac{n}{2} \cdot (n - 6) \\
&\geq \frac{n}{2} \cdot \frac{n}{2} && (\text{para } n \geq 12) \\
&= c \cdot g(n). && \left(\text{para } c = \frac{1}{4}\right)
\end{aligned}
$$

Notação Ω: exemplo

Exemplo

$\frac{1}{2}n^2 - 3n \in \Omega(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$
\begin{aligned}
f(n) &= \frac{1}{2}n^2 - 3n \\
&= \frac{n}{2} \cdot (n - 6) \\
&\geq \frac{n}{2} \cdot \frac{n}{2} && (\text{para } n \geq 12) \\
&= c \cdot g(n). && (\text{para } c = \frac{1}{4})
\end{aligned}
$$

Notação Ω: exemplo

Exemplo

$\frac{1}{2}n^2 - 3n \in \Omega(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$
\begin{aligned}
f(n) &= \frac{1}{2}n^2 - 3n \\
&= \frac{n}{2} \cdot (n - 6) \\
&\geq \frac{n}{2} \cdot \frac{n}{2} && \text{(para } n \geq 12) \\
&= c \cdot g(n). && \left(\text{para } c = \frac{1}{4}\right)
\end{aligned}
$$

Notação Ω: exemplo

Exemplo

$\frac{1}{2}n^2 - 3n \in \Omega(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Note que

$$
\begin{aligned}
f(n) &= \frac{1}{2}n^2 - 3n \\
&= \frac{n}{2} \cdot (n-6) \\
&\geq \frac{n}{2} \cdot \frac{n}{2} && (\text{para } n \geq 12) \\
&= c \cdot g(n). && (\text{para } c = \frac{1}{4})
\end{aligned}
$$

Notação Ω: exemplo

Exemplo

$4n^2 - 100n - 500 = \Omega(n^2)$

- Temos $f(n) = 4n^2 - 100n - 500$ e $g(n) = n^2$.
- Precisamos encontrar constantes positivas c e n_0 tais que $cn^2 \leq 4n^2 - 100n - 500$ para todo $n \geq n_0$.
- Escolhendo $c = 3$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 104,7$.
- Escolhendo $c = 2$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 54,5$.
- Escolhendo $c = 1$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 37,7$.

Notação Ω: exemplo

Exemplo

$4n^2 - 100n - 500 = \Omega(n^2)$

- Temos $f(n) = 4n^2 - 100n - 500$ e $g(n) = n^2$.
- Precisamos encontrar constantes positivas c e n_0 tais que $cn^2 \leq 4n^2 - 100n - 500$ para todo $n \geq n_0$.
- Escolhendo $c = 3$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 104,7$.
- Escolhendo $c = 2$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 54,5$.
- Escolhendo $c = 1$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 37,7$.

Notação Ω: exemplo

Exemplo

$4n^2 - 100n - 500 = \Omega(n^2)$

- Temos $f(n) = 4n^2 - 100n - 500$ e $g(n) = n^2$.
- Precisamos encontrar constantes positivas c e n_0 tais que $cn^2 \leq 4n^2 - 100n - 500$ para todo $n \geq n_0$.
- Escolhendo $c = 3$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 104{,}7$.
- Escolhendo $c = 2$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 54{,}5$.
- Escolhendo $c = 1$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 37{,}7$.

Notação Ω: exemplo

Exemplo

$4n^2 - 100n - 500 = \Omega(n^2)$

- Temos $f(n) = 4n^2 - 100n - 500$ e $g(n) = n^2$.
- Precisamos encontrar constantes positivas c e n_0 tais que $cn^2 \leq 4n^2 - 100n - 500$ para todo $n \geq n_0$.
- Escolhendo $c = 3$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 104{,}7$.
- Escolhendo $c = 2$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 54{,}5$.
- Escolhendo $c = 1$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 37{,}7$.

Notação Ω: exemplo

Exemplo

$4n^2 - 100n - 500 = \Omega(n^2)$

- Temos $f(n) = 4n^2 - 100n - 500$ e $g(n) = n^2$.
- Precisamos encontrar constantes positivas c e n_0 tais que $cn^2 \leq 4n^2 - 100n - 500$ para todo $n \geq n_0$.
- Escolhendo $c = 3$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 104,7$.
- Escolhendo $c = 2$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 54,5$.
- Escolhendo $c = 1$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 37,7$.

Notação Ω: exemplo

Exemplo

$4n^2 - 100n - 500 = \Omega(n^2)$

- Temos $f(n) = 4n^2 - 100n - 500$ e $g(n) = n^2$.
- Precisamos encontrar constantes positivas c e n_0 tais que $cn^2 \leq 4n^2 - 100n - 500$ para todo $n \geq n_0$.
- Escolhendo $c = 3$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 104,7$.
- Escolhendo $c = 2$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 54,5$.
- Escolhendo $c = 1$, obtemos que $cn^2 \leq 4n^2 - 100n - 500$ é verdadeira para todo $n \geq n_0 = 37,7$.

Notação Θ

Definição

Dada uma função $g(n)$, a classe $\Theta(g(n))$ é o conjunto de funções $f(n)$ tais que:

- existem constantes positivas c_1, c_2 e n_0 de modo que

$$0 \leq c_1 \, g(n) \leq f(n) \leq c_2 \, g(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce tão rápido quanto $g(n)$.

Notação Θ

Definição

Dada uma função $g(n)$, a classe $\Theta(g(n))$ é o conjunto de funções $f(n)$ tais que:

- existem constantes positivas c_1, c_2 e n_0 de modo que

$$0 \leq c_1\, g(n) \leq f(n) \leq c_2\, g(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce tão rápido quanto $g(n)$.

Notação Θ

Definição

Dada uma função $g(n)$, a classe $\Theta(g(n))$ é o conjunto de funções $f(n)$ tais que:

- existem constantes positivas c_1, c_2 e n_0 de modo que

$$0 \leq c_1\, g(n) \leq f(n) \leq c_2\, g(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce tão rápido quanto $g(n)$.

Notação Θ

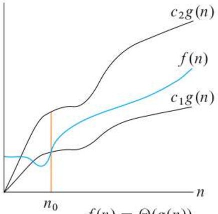
Definição

Dada uma função $g(n)$, a classe $\Theta(g(n))$ é o conjunto de funções $f(n)$ tais que:

- existem constantes positivas c_1, c_2 e n_0 de modo que

$$0 \leq c_1\, g(n) \leq f(n) \leq c_2\, g(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce tão rápido quanto $g(n)$.



$f(n) = \Theta(g(n))$

Notação Θ

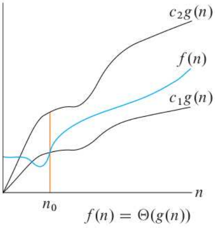
Definição

Dada uma função $g(n)$, a classe $\Theta(g(n))$ é o conjunto de funções $f(n)$ tais que:

- existem constantes positivas c_1, c_2 e n_0 de modo que

$$0 \leq c_1\, g(n) \leq f(n) \leq c_2\, g(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce tão rápido quanto $g(n)$.



Escrevemos $f(n) = \Theta(g(n))$ quando $f(n) \in \Theta(g(n))$.

Notação Θ

Teorema

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

Exercício: Prove este teorema.

Notação Θ

Teorema

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

Exercício: Prove este teorema.

Notação Θ: exemplo

Exemplo

$\frac{1}{2}n^2 - 3n = \Theta(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Escolha valores

$$c_1 = \frac{1}{4}, \quad c_2 = \frac{1}{2} \quad \text{e} \quad n_0 = 12.$$

- Então, supondo $n \geq n_0$, já verificamos que

$$c_1\, g(n) \leq f(n) \leq c_2\, g(n).$$

Notação Θ: exemplo

Exemplo

$\frac{1}{2}n^2 - 3n = \Theta(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Escolha valores

$$c_1 = \frac{1}{4}, \quad c_2 = \frac{1}{2} \quad \text{e} \quad n_0 = 12.$$

- Então, supondo $n \geq n_0$, já verificamos que

$$c_1\, g(n) \leq f(n) \leq c_2\, g(n).$$

Notação Θ: exemplo

Exemplo

$\frac{1}{2}n^2 - 3n = \Theta(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Escolha valores

$$c_1 = \frac{1}{4}, \quad c_2 = \frac{1}{2} \quad \text{e} \quad n_0 = 12.$$

- Então, supondo $n \geq n_0$, já verificamos que

$$c_1\, g(n) \leq f(n) \leq c_2\, g(n).$$

Notação Θ: exemplo

Exemplo

$\frac{1}{2}n^2 - 3n = \Theta(n^2)$

- Temos $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$.
- Escolha valores

$$c_1 = \frac{1}{4}, \quad c_2 = \frac{1}{2} \quad \text{e} \quad n_0 = 12.$$

- Então, supondo $n \geq n_0$, já verificamos que

$$c_1\, g(n) \leq f(n) \leq c_2\, g(n).$$

Obtendo Θ através de limites

Teorema

Sejam $f(n)$ e $g(n)$ funções não negativas tais que

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = c$$

para alguma constante $c > 0$. Então, $f(n) = \Theta(g(n))$.

Prova: Usaremos o fato de que o limite existe e é positivo para mostrar que $f(n) = O(g(n))$ e $f(n) = \Omega(g(n))$, como requerido pela definição de Θ. Como $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c > 0$, segue da definição de limite que existe alguma constante n_0 tal que para todo $n \geq n_0$ a razão $f(n)/g(n)$ está sempre entre $\frac{1}{2}c$ e $2c$. Assim, $f(n) \leq 2cg(n)$ para todo $n \geq n_0$, o que implica $f(n) = O(g(n))$; e $f(n) \geq \frac{1}{2}cg(n)$ para todo $n \geq n_0$, o que implica $f(n) = \Omega(g(n))$. □

Obtendo Θ através de limites

Teorema

Sejam $f(n)$ e $g(n)$ funções não negativas tais que

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$$

para alguma constante $c > 0$. Então, $f(n) = \Theta(g(n))$.

Prova: Usaremos o fato de que o limite existe e é positivo para mostrar que $f(n) = O(g(n))$ e $f(n) = \Omega(g(n))$, como requerido pela definição de Θ. Como $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c > 0$, segue da definição de limite que existe alguma constante n_0 tal que para todo $n \geq n_0$ a razão $f(n)/g(n)$ está sempre entre $\frac{1}{2}c$ e $2c$. Assim, $f(n) \leq 2cg(n)$ para todo $n \geq n_0$, o que implica $f(n) = O(g(n))$; e $f(n) \geq \frac{1}{2}cg(n)$ para todo $n \geq n_0$, o que implica $f(n) = \Omega(g(n))$. $\square$

Exercício

Exercício: Suponha $f(n) = 5n\lg n + 8\lg^2 n - 11$ e $g(n) = n\lg n$. Mostre que $f(n) = \Theta(g(n))$.

Solução: usando limites:

$$
\begin{aligned}
&\lim_{n\to\infty} \frac{5n\lg n + 8\lg^2 n - 11}{n\lg n} \\
&= \lim_{n\to\infty} 5 + 8\frac{\lg n}{n} - 11\frac{1}{n\lg n} \\
&= 5 + 8\cdot 0 - 11\cdot 0 \\
&= 5.
\end{aligned}
$$

Logo, como o limite existe, então $f(n) = \Theta(g(n))$.

Exercício

Exercício: Suponha $f(n) = 5n\lg n + 8\lg^2 n - 11$ e $g(n) = n\lg n$. Mostre que $f(n) = \Theta(g(n))$.

Solução: usando limites:

$$
\begin{aligned}
&\lim_{n\to\infty} \frac{5n\lg n + 8\lg^2 n - 11}{n\lg n} \\
&= \lim_{n\to\infty} 5 + 8\frac{\lg n}{n} - 11\frac{1}{n\lg n} \\
&= 5 + 8\cdot 0 - 11\cdot 0 \\
&= 5.
\end{aligned}
$$

Logo, como o limite existe, então $f(n) = \Theta(g(n))$.

Notação o

Definição

Dada função $g(n)$, a classe $o(g(n))$ é o conjunto de funções $f(n)$ tais que:

- para toda constante $c > 0$, existe um número n_0
- tal que $0 \leq f(n) < cg(n)$ para todo $n \geq n_0$

Dizemos que $f(n)$ cresce mais lentamente que $g(n)$.

Notação o

Definição

Dada função $g(n)$, a classe $o(g(n))$ é o conjunto de funções $f(n)$ tais que:

- para toda constante $c > 0$, existe um número n_0
- tal que $0 \leq f(n) < cg(n)$ para todo $n \geq n_0$

Dizemos que $f(n)$ cresce mais lentamente que $g(n)$.

Notação o

Definição

Dada função $g(n)$, a classe $o(g(n))$ é o conjunto de funções $f(n)$ tais que:

- para **toda** constante $c > 0$, existe um número n_0
- tal que $0 \leq f(n) < cg(n)$ para todo $n \geq n_0$

Dizemos que $f(n)$ cresce **mais lentamente** que $g(n)$.

Notação o

Definição

Dada função $g(n)$, a classe $o(g(n))$ é o conjunto de funções $f(n)$ tais que:

- para toda constante $c > 0$, existe um número n_0
- tal que $0 \leq f(n) < cg(n)$ para todo $n \geq n_0$

Dizemos que $f(n)$ cresce mais lentamente que $g(n)$.

Notação o: exemplo

Exemplo

$1000n^2 = o(n^3)$

- Temos $f(n) = 1000n^2$ e $g(n) = n^3$.
- Seja $c > 0$ uma constante arbitrária
- Devemos mostrar que existe constante positiva n_0 tal que $1000n^2 < cn^3$ para todo $n \geq n_0$.
- Assim, temos

$$1000n^2 < cn^3 \iff \frac{1000n^2}{cn^2} < \frac{cn^3}{cn^2} \iff \frac{1000}{c} < n$$

- Portanto, obtemos $n_0 = \left\lceil \frac{1000}{c} \right\rceil + 1$.

Notação o: exemplo

Exemplo

$1000n^2 = o(n^3)$

- Temos $f(n) = 1000n^2$ e $g(n) = n^3$.
- Seja $c > 0$ uma constante arbitrária
- Devemos mostrar que existe constante positiva n_0 tal que $1000n^2 < cn^3$ para todo $n \geq n_0$.
- Assim, temos

$$1000n^2 < cn^3 \iff \frac{1000n^2}{cn^2} < \frac{cn^3}{cn^2} \iff \frac{1000}{c} < n$$

- Portanto, obtemos $n_0 = \left\lceil \frac{1000}{c} \right\rceil + 1$.

Notação o: exemplo

Exemplo

$1000n^2 = o(n^3)$

- Temos $f(n) = 1000n^2$ e $g(n) = n^3$.
- Seja $c > 0$ uma constante arbitrária
- Devemos mostrar que existe constante positiva n_0 tal que $1000n^2 < cn^3$ para todo $n \geq n_0$.
- Assim, temos

$$1000n^2 < cn^3 \iff \frac{1000n^2}{cn^2} < \frac{cn^3}{cn^2} \iff \frac{1000}{c} < n$$

- Portanto, obtemos $n_0 = \left\lceil \frac{1000}{c} \right\rceil + 1$.

Notação o: exemplo

Exemplo

$1000n^2 = o(n^3)$

- Temos $f(n) = 1000n^2$ e $g(n) = n^3$.
- Seja $c > 0$ uma constante arbitrária
- Devemos mostrar que existe constante positiva n_0 tal que $1000n^2 < cn^3$ para todo $n \geq n_0$.
- Assim, temos

$$1000n^2 < cn^3 \iff \frac{1000n^2}{cn^2} < \frac{cn^3}{cn^2} \iff \frac{1000}{c} < n$$

- Portanto, obtemos $n_0 = \left\lceil \frac{1000}{c} \right\rceil + 1$.

Notação o: exemplo

Exemplo

$1000n^2 = o(n^3)$

- Temos $f(n) = 1000n^2$ e $g(n) = n^3$.
- Seja $c > 0$ uma constante arbitrária
- Devemos mostrar que existe constante positiva n_0 tal que $1000n^2 < cn^3$ para todo $n \geq n_0$.
- Assim, temos

$$1000n^2 < cn^3 \iff \frac{1000n^2}{cn^2} < \frac{cn^3}{cn^2} \iff \frac{1000}{c} < n$$

- Portanto, obtemos $n_0 = \left\lceil \frac{1000}{c} \right\rceil + 1$.

Notação o: exemplo

Exemplo

$1000n^2 = o(n^3)$

- Temos $f(n) = 1000n^2$ e $g(n) = n^3$.
- Seja $c > 0$ uma constante arbitrária
- Devemos mostrar que existe constante positiva n_0 tal que $1000n^2 < cn^3$ para todo $n \geq n_0$.
- Assim, temos

$$1000n^2 < cn^3 \iff \frac{1000n^2}{cn^2} < \frac{cn^3}{cn^2} \iff \frac{1000}{c} < n$$

- Portanto, obtemos $n_0 = \left\lceil \frac{1000}{c} \right\rceil + 1$.

Notação ω

Definição

Dada função $g(n)$, a classe $\omega(g(n))$ é o conjunto de funções $f(n)$ tais que:

- para toda constante $c > 0$, existe um número n_0 tal que

$$0 \leq cg(n) < f(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce mais rapidamente que $g(n)$.

Notação ω

Definição

Dada função $g(n)$, a classe $\omega(g(n))$ é o conjunto de funções $f(n)$ tais que:

- para toda constante $c > 0$, existe um número n_0 tal que
$$0 \leq cg(n) < f(n) \text{ para todo } n \geq n_0$$

Dizemos que $f(n)$ cresce mais rapidamente que $g(n)$.

Notação ω

Definição

Dada função $g(n)$, a classe $\omega(g(n))$ é o conjunto de funções $f(n)$ tais que:

- para toda constante $c > 0$, existe um número n_0 tal que

$$0 \le cg(n) < f(n) \text{ para todo } n \ge n_0$$

Dizemos que $f(n)$ cresce mais rapidamente que $g(n)$.

Notação ω: exemplo

Exemplo

$\frac{1}{1000}n^2 = \omega(n)$

- Temos $f(n) = \frac{n^2}{1000}$ e $g(n) = n$.
- Seja $c > 0$ uma constante arbitrária.
- Devemos encontrar constante positiva n_0 tal que $\frac{n^2}{1000} > cn$ para todo $n \geq n_0$.
- Dividindo ambos os lados por n, obtemos

$$\frac{n}{1000} > c \iff n > 1000c$$

- Portanto, $n_0 = \lceil 1000c \rceil + 1$.

Notação ω: exemplo

Exemplo

$\frac{1}{1000}n^2 = \omega(n)$

- Temos $f(n) = \frac{n^2}{1000}$ e $g(n) = n$.
- Seja $c > 0$ uma constante arbitrária.
- Devemos encontrar constante positiva n_0 tal que $\frac{n^2}{1000} > cn$ para todo $n \geq n_0$.
- Dividindo ambos os lados por n, obtemos

$$\frac{n}{1000} > c \iff n > 1000c$$

- Portanto, $n_0 = \lceil 1000c \rceil + 1$.

Notação ω: exemplo

Exemplo

$\frac{1}{1000}n^2 = \omega(n)$

- Temos $f(n) = \frac{n^2}{1000}$ e $g(n) = n$.
- Seja $c > 0$ uma constante arbitrária.
- Devemos encontrar constante positiva n_0 tal que $\frac{n^2}{1000} > cn$ para todo $n \geq n_0$.
- Dividindo ambos os lados por n, obtemos

$$\frac{n}{1000} > c \iff n > 1000c$$

- Portanto, $n_0 = \lceil 1000c \rceil + 1$.

Notação ω: exemplo

Exemplo

$\frac{1}{1000}n^2 = \omega(n)$

- Temos $f(n) = \frac{n^2}{1000}$ e $g(n) = n$.
- Seja $c > 0$ uma constante arbitrária.
- Devemos encontrar constante positiva n_0 tal que $\frac{n^2}{1000} > cn$ para todo $n \geq n_0$.
- Dividindo ambos os lados por n, obtemos

$$\frac{n}{1000} > c \iff n > 1000c$$

- Portanto, $n_0 = \lceil 1000c \rceil + 1$.

Notação ω: exemplo

Exemplo

$\frac{1}{1000}n^2 = \omega(n)$

- Temos $f(n) = \frac{n^2}{1000}$ e $g(n) = n$.
- Seja $c > 0$ uma constante arbitrária.
- Devemos encontrar constante positiva n_0 tal que $\frac{n^2}{1000} > cn$ para todo $n \geq n_0$.
- Dividindo ambos os lados por n, obtemos

$$\frac{n}{1000} > c \iff n > 1000c$$

- Portanto, $n_0 = \lceil 1000c \rceil + 1$.

Notação ω: exemplo

Exemplo

$\frac{1}{1000}n^2 = \omega(n)$

- Temos $f(n) = \frac{n^2}{1000}$ e $g(n) = n$.
- Seja $c > 0$ uma constante arbitrária.
- Devemos encontrar constante positiva n_0 tal que $\frac{n^2}{1000} > cn$ para todo $n \geq n_0$.
- Dividindo ambos os lados por n, obtemos

$$\frac{n}{1000} > c \iff n > 1000c$$

- Portanto, $n_0 = \lceil 1000c \rceil + 1$.

Equivalências

Obtendo o e ω através de limites.

Condições equivalentes

Sejam $f(n)$ e $g(n)$ funções não negativas, então:

- $f(n) \in o(g(n))$ se, e somente se, $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$.
- $f(n) \in \omega(g(n))$ se, e somente se, $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$.

Equivalências

Obtendo o e ω através de limites.

Condições equivalentes

Sejam $f(n)$ e $g(n)$ funções não negativas, então:

- $f(n) \in o(g(n))$ se, e somente se, $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$.
- $f(n) \in \omega(g(n))$ se, e somente se, $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$.

Equivalências

Obtendo o e ω através de limites.

Condições equivalentes

Sejam $f(n)$ e $g(n)$ funções não negativas, então:

- $f(n) \in o(g(n))$ se, e somente se, $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$.
- $f(n) \in \omega(g(n))$ se, e somente se, $\lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty$.

Notação assintótica e crescimento de funções

- Propriedades das notações assintóticas

Propriedades das classes

Transitividade

- Se $f(n) = O(g(n))$ e $g(n) = O(h(n))$, então $f(n) = O(h(n))$.
- Se $f(n) = \Omega(g(n))$ e $g(n) = \Omega(h(n))$, então $f(n) = \Omega(h(n))$.
- Se $f(n) = \Theta(g(n))$ e $g(n) = \Theta(h(n))$, então $f(n) = \Theta(h(n))$.
- Se $f(n) = o(g(n))$ e $g(n) = o(h(n))$, então $f(n) = o(h(n))$.
- Se $f(n) = \omega(g(n))$ e $g(n) = \omega(h(n))$, então $f(n) = \omega(h(n))$.

Propriedades das classes

Transitividade

- Se $f(n) = O(g(n))$ e $g(n) = O(h(n))$, então $f(n) = O(h(n))$.
- Se $f(n) = \Omega(g(n))$ e $g(n) = \Omega(h(n))$, então $f(n) = \Omega(h(n))$.
- Se $f(n) = \Theta(g(n))$ e $g(n) = \Theta(h(n))$, então $f(n) = \Theta(h(n))$.
- Se $f(n) = o(g(n))$ e $g(n) = o(h(n))$, então $f(n) = o(h(n))$.
- Se $f(n) = \omega(g(n))$ e $g(n) = \omega(h(n))$, então $f(n) = \omega(h(n))$.

Propriedades das classes

Transitividade

- Se $f(n) = O(g(n))$ e $g(n) = O(h(n))$, então $f(n) = O(h(n))$.
- Se $f(n) = \Omega(g(n))$ e $g(n) = \Omega(h(n))$, então $f(n) = \Omega(h(n))$.
- Se $f(n) = \Theta(g(n))$ e $g(n) = \Theta(h(n))$, então $f(n) = \Theta(h(n))$.
- Se $f(n) = o(g(n))$ e $g(n) = o(h(n))$, então $f(n) = o(h(n))$.
- Se $f(n) = \omega(g(n))$ e $g(n) = \omega(h(n))$, então $f(n) = \omega(h(n))$.

Propriedades das classes

Transitividade

- Se $f(n) = O(g(n))$ e $g(n) = O(h(n))$, então $f(n) = O(h(n))$.
- Se $f(n) = \Omega(g(n))$ e $g(n) = \Omega(h(n))$, então $f(n) = \Omega(h(n))$.
- Se $f(n) = \Theta(g(n))$ e $g(n) = \Theta(h(n))$, então $f(n) = \Theta(h(n))$.
- Se $f(n) = o(g(n))$ e $g(n) = o(h(n))$, então $f(n) = o(h(n))$.
- Se $f(n) = \omega(g(n))$ e $g(n) = \omega(h(n))$, então $f(n) = \omega(h(n))$.

Propriedades das classes

Transitividade

- Se $f(n) = O(g(n))$ e $g(n) = O(h(n))$, então $f(n) = O(h(n))$.
- Se $f(n) = \Omega(g(n))$ e $g(n) = \Omega(h(n))$, então $f(n) = \Omega(h(n))$.
- Se $f(n) = \Theta(g(n))$ e $g(n) = \Theta(h(n))$, então $f(n) = \Theta(h(n))$.
- Se $f(n) = o(g(n))$ e $g(n) = o(h(n))$, então $f(n) = o(h(n))$.
- Se $f(n) = \omega(g(n))$ e $g(n) = \omega(h(n))$, então $f(n) = \omega(h(n))$.

Propriedades das classes

Transitividade

- Se $f(n) = O(g(n))$ e $g(n) = O(h(n))$, então $f(n) = O(h(n))$.
- Se $f(n) = \Omega(g(n))$ e $g(n) = \Omega(h(n))$, então $f(n) = \Omega(h(n))$.
- Se $f(n) = \Theta(g(n))$ e $g(n) = \Theta(h(n))$, então $f(n) = \Theta(h(n))$.
- Se $f(n) = o(g(n))$ e $g(n) = o(h(n))$, então $f(n) = o(h(n))$.
- Se $f(n) = \omega(g(n))$ e $g(n) = \omega(h(n))$, então $f(n) = \omega(h(n))$.

Propriedades das classes

Reflexividade

- $f(n) = O(f(n))$.
- $f(n) = \Omega(f(n))$.
- $f(n) = \Theta(f(n))$.

Simetria

- $f(n) = \Theta(g(n))$ se, e somente se, $g(n) = \Theta(f(n))$.

Simetria Transposta

- $f(n) = O(g(n))$ se, e somente se, $g(n) = \Omega(f(n))$.
- $f(n) = o(g(n))$ se, e somente se, $g(n) = \omega(f(n))$.

Propriedades das classes

Reflexividade

- $f(n) = O(f(n))$.
- $f(n) = \Omega(f(n))$.
- $f(n) = \Theta(f(n))$.

Simetria

- $f(n) = \Theta(g(n))$ se, e somente se, $g(n) = \Theta(f(n))$.

Simetria Transposta

- $f(n) = O(g(n))$ se, e somente se, $g(n) = \Omega(f(n))$.
- $f(n) = o(g(n))$ se, e somente se, $g(n) = \omega(f(n))$.

Propriedades das classes

Reflexividade

- $f(n) = O(f(n))$.
- $f(n) = \Omega(f(n))$.
- $f(n) = \Theta(f(n))$.

Simetria

- $f(n) = \Theta(g(n))$ se, e somente se, $g(n) = \Theta(f(n))$.

Simetria Transposta

- $f(n) = O(g(n))$ se, e somente se, $g(n) = \Omega(f(n))$.
- $f(n) = o(g(n))$ se, e somente se, $g(n) = \omega(f(n))$.

Propriedades das classes

Reflexividade

- $f(n) = O(f(n))$.
- $f(n) = \Omega(f(n))$.
- $f(n) = \Theta(f(n))$.

Simetria

- $f(n) = \Theta(g(n))$ se, e somente se, $g(n) = \Theta(f(n))$.

Simetria Transposta

- $f(n) = O(g(n))$ se, e somente se, $g(n) = \Omega(f(n))$.
- $f(n) = o(g(n))$ se, e somente se, $g(n) = \omega(f(n))$.

Propriedades das classes

Reflexividade

- $f(n) = O(f(n))$.
- $f(n) = \Omega(f(n))$.
- $f(n) = \Theta(f(n))$.

Simetria

- $f(n) = \Theta(g(n))$ se, e somente se, $g(n) = \Theta(f(n))$.

Simetria Transposta

- $f(n) = O(g(n))$ se, e somente se, $g(n) = \Omega(f(n))$.
- $f(n) = o(g(n))$ se, e somente se, $g(n) = \omega(f(n))$.

Propriedades das classes

Reflexividade

- $f(n) = O(f(n))$.
- $f(n) = \Omega(f(n))$.
- $f(n) = \Theta(f(n))$.

Simetria

- $f(n) = \Theta(g(n))$ se, e somente se, $g(n) = \Theta(f(n))$.

Simetria Transposta

- $f(n) = O(g(n))$ se, e somente se, $g(n) = \Omega(f(n))$.
- $f(n) = o(g(n))$ se, e somente se, $g(n) = \omega(f(n))$.

Propriedades das classes

Reflexividade

- $f(n) = O(f(n))$.
- $f(n) = \Omega(f(n))$.
- $f(n) = \Theta(f(n))$.

Simetria

- $f(n) = \Theta(g(n))$ se, e somente se, $g(n) = \Theta(f(n))$.

Simetria Transposta

- $f(n) = O(g(n))$ se, e somente se, $g(n) = \Omega(f(n))$.
- $f(n) = o(g(n))$ se, e somente se, $g(n) = \omega(f(n))$.

Propriedades das classes

Reflexividade

- $f(n) = O(f(n))$.
- $f(n) = \Omega(f(n))$.
- $f(n) = \Theta(f(n))$.

Simetria

- $f(n) = \Theta(g(n))$ se, e somente se, $g(n) = \Theta(f(n))$.

Simetria Transposta

- $f(n) = O(g(n))$ se, e somente se, $g(n) = \Omega(f(n))$.
- $f(n) = o(g(n))$ se, e somente se, $g(n) = \omega(f(n))$.

Propriedades das classes

Reflexividade

- $f(n) = O(f(n))$.
- $f(n) = \Omega(f(n))$.
- $f(n) = \Theta(f(n))$.

Simetria

- $f(n) = \Theta(g(n))$ se, e somente se, $g(n) = \Theta(f(n))$.

Simetria Transposta

- $f(n) = O(g(n))$ se, e somente se, $g(n) = \Omega(f(n))$.
- $f(n) = o(g(n))$ se, e somente se, $g(n) = \omega(f(n))$.

Limites superiores para algumas funções comuns

Limites superiores para algumas funções comuns

- Polinômios

Polinômios

Um polinômio é uma função que pode ser escrita na forma $f(n) = a_0 + a_1 n + a_2 n^2 + \cdots + a_d n^d$ para alguma constante $d > 0$, tal que $a_d \neq 0$. O valor d é chamado grau do polinômio.

- **Exemplo:** Funções da forma $pn^2 + qn + r$, com $p \neq 0$, são polinômios de grau 2.

Polinômios

Teorema

Seja $f(n)$ um polinômio de grau d com coeficiente $a_d > 0$. Então $f(n) = O(n^d)$.

Prova:

- Considere $f(n) = a_0 + a_1 n + \cdots + a_d n^d$, com $a_d > 0$. O limitante superior é uma consequência imediata de um teorema anterior.
- Primeiro, note que os coeficientes a_j para $j < d$ podem ser negativos, mas, de qualquer modo, $a_j n^j \leq |a_j| n^d$ para todo $n \geq 1$. Assim, cada termo no polinômio é $O(n^d)$.
- Como $f(n)$ é uma soma de um número constante de funções, cada uma das quais é $O(n^d)$, segue por um teorema anterior que $f(n)$ é $O(n^d)$. □

Polinômios

Teorema

Seja $f(n)$ um polinômio de grau d com coeficiente $a_d > 0$. Então $f(n) = O(n^d)$.

Prova:

- Considere $f(n) = a_0 + a_1 n + \cdots + a_d n^d$, com $a_d > 0$. O limitante superior é uma consequência imediata de um teorema anterior.
- Primeiro, note que os coeficientes a_j para $j < d$ podem ser negativos, mas, de qualquer modo, $a_j n^j \leq |a_j| n^d$ para todo $n \geq 1$. Assim, cada termo no polinômio é $O(n^d)$.
- Como $f(n)$ é uma soma de um número constante de funções, cada uma das quais é $O(n^d)$, segue por um teorema anterior que $f(n)$ é $O(n^d)$. $\square$

Polinômios

Teorema

Seja $f(n)$ um polinômio de grau d com coeficiente $a_d > 0$. Então $f(n) = O(n^d)$.

Prova:

- Considere $f(n) = a_0 + a_1 n + \cdots + a_d n^d$, com $a_d > 0$. O limitante superior é uma consequência imediata de um teorema anterior.
- Primeiro, note que os coeficientes a_j para $j < d$ podem ser negativos, mas, de qualquer modo, $a_j n^j \leq |a_j| n^d$ para todo $n \geq 1$. Assim, cada termo no polinômio é $O(n^d)$.
- Como $f(n)$ é uma soma de um número constante de funções, cada uma das quais é $O(n^d)$, segue por um teorema anterior que $f(n)$ é $O(n^d)$. $\square$

Polinômios

Teorema

Seja $f(n)$ um polinômio de grau d com coeficiente $a_d > 0$. Então $f(n) = O(n^d)$.

Prova:

- Considere $f(n) = a_0 + a_1 n + \cdots + a_d n^d$, com $a_d > 0$. O limitante superior é uma consequência imediata de um teorema anterior.
- Primeiro, note que os coeficientes a_j para $j < d$ podem ser negativos, mas, de qualquer modo, $a_j n^j \leq |a_j| n^d$ para todo $n \geq 1$. Assim, cada termo no polinômio é $O(n^d)$.
- Como $f(n)$ é uma soma de um número constante de funções, cada uma das quais é $O(n^d)$, segue por um teorema anterior que $f(n)$ é $O(n^d)$. $\square$

Polinômios

Um algoritmo de tempo polinomial é aquele cujo tempo de execução $T(n)$ é $O(n^d)$ para alguma constante d, onde d é independente do tamanho da entrada.

Exemplos:

- $O(n^2)$
- $O(n^3)$
- $O(\sqrt{n}) = O(n^{1/2})$
- $O(n^{1.59})$

Convencionou-se que algoritmos eficientes são os algoritmos de tempo polinomial.

Polinômios

Um algoritmo de tempo polinomial é aquele cujo tempo de execução $T(n)$ é $O(n^d)$ para alguma constante d, onde d é independente do tamanho da entrada.

Exemplos:

- $O(n^2)$
- $O(n^3)$
- $O(\sqrt{n}) = O(n^{1/2})$
- $O(n^{1.59})$

Convencionou-se que algoritmos eficientes são os algoritmos de tempo polinomial.

Limites superiores para algumas funções comuns

- Logaritmos

Logaritmos

Se $n, b \in \mathbb{R}$, com $0 < b \neq 1$ e $n > 0$, então:

$$\log_b n = x \iff b^x = n.$$

- Uma maneira de obter uma noção aproximada de quão rápido $\log_b n$ cresce é observar que, se o arredondarmos para o inteiro mais próximo, ele será um a menos que o número de dígitos na representação de base b do número n. (Assim, por exemplo, $1 + \lfloor \log_2 n \rfloor$ é o número de bits necessários para representar n.)

Teorema

Para todo $b > 1$ e todo $x > 0$, temos que $\log_b n = O(n^x)$. □

Logaritmos

Se $n, b \in \mathbb{R}$, com $0 < b \neq 1$ e $n > 0$, então:

$$\log_b n = x \iff b^x = n.$$

- Uma maneira de obter uma noção aproximada de quão rápido $\log_b n$ cresce é observar que, se o arredondarmos para o inteiro mais próximo, ele será um a menos que o número de dígitos na representação de base b do número n. (Assim, por exemplo, $1 + \lfloor \log_2 n \rfloor$ é o número de bits necessários para representar n.)

Teorema

Para todo $b > 1$ e todo $x > 0$, temos que $\log_b n = O(n^x)$. $\square$

Logaritmos

Se $n, b \in \mathbb{R}$, com $0 < b \neq 1$ e $n > 0$, então:

$$\log_b n = x \iff b^x = n.$$

- Uma maneira de obter uma noção aproximada de quão rápido $\log_b n$ cresce é observar que, se o arredondarmos para o inteiro mais próximo, ele será um a menos que o número de dígitos na representação de base b do número n. (Assim, por exemplo, $1 + \lfloor \log_2 n \rfloor$ é o número de bits necessários para representar n.)

Teorema

Para todo $b > 1$ e todo $x > 0$, temos que $\log_b n = O(n^x)$. $\square$

Logaritmos

Pode-se traduzir diretamente entre logaritmos de diferentes bases usando a seguinte identidade fundamental:

$$\log_a n = \frac{\log_b n}{\log_b a}$$

- Esta equação justifica porque frequentemente os livros escrevem $O(\log n)$ ou $O(\lg n)$ sem indicar a base do logaritmo.

Teorema

Para $a, b > 1$ e todo $n > 0$, temos que $\log_a n = \Theta(\log_b n)$. $\square$

Logaritmos

Pode-se traduzir diretamente entre logaritmos de diferentes bases usando a seguinte identidade fundamental:

$$\log_a n = \frac{\log_b n}{\log_b a}$$

- Esta equação justifica porque frequentemente os livros escrevem $O(\log n)$ ou $O(\lg n)$ sem indicar a base do logaritmo.

Teorema

Para $a, b > 1$ e todo $n > 0$, temos que $\log_a n = \Theta(\log_b n)$. □

Logaritmos

Pode-se traduzir diretamente entre logaritmos de diferentes bases usando a seguinte identidade fundamental:

$$\log_a n = \frac{\log_b n}{\log_b a}$$

- Esta equação justifica porque frequentemente os livros escrevem $O(\log n)$ ou $O(\lg n)$ sem indicar a base do logaritmo.

Teorema

Para $a, b > 1$ e todo $n > 0$, temos que $\log_a n = \Theta(\log_b n)$. $\square$

Limites superiores para algumas funções comuns

- Exponenciais

Exponenciais

Funções exponenciais são funções da forma $f(n) = a^n$ para alguma constante base a. Aqui, estaremos preocupados com o caso no qual $a > 1$, o que resulta numa função que cresce rapidamente.

Teorema

Para toda constante $d > 0$ e $a > 1$, temos que $n^d = o(a^n)$.

Em outras palavras, uma função exponencial cresce mais rapidamente que uma função polinomial.

Exponenciais

Funções exponenciais são funções da forma $f(n) = a^n$ para alguma constante base a. Aqui, estaremos preocupados com o caso no qual $a > 1$, o que resulta numa função que cresce rapidamente.

Teorema

Para toda constante $d > 0$ e $a > 1$, temos que $n^d = o(a^n)$.

Em outras palavras, uma função exponencial cresce mais rapidamente que uma função polinomial.

Funções importantes

Nomenclatura e consumo de tempo

- $O(1)$: tempo constante
  - não depende de n, operações executadas um número fixo de vezes.
- $O(\lg n)$: logarítmico
  - lg indica $\log_2$
  - típica de algoritmos que resolvem um problema transformando-o em problemas menores.
  - quando n dobra, o tempo aumenta em uma constante
  - Ex: Busca binária
  - Outros exemplos durante o curso

Nomenclatura e consumo de tempo

- $O(1)$: tempo constante
  - não depende de n, operações executadas um número fixo de vezes.
- $O(\lg n)$: logarítmico
  - $\lg$ indica $\log_2$
  - típica de algoritmos que resolvem um problema transformando-o em problemas menores.
  - quando n dobra, o tempo aumenta em uma constante
  - Ex: Busca binária
  - Outros exemplos durante o curso

Nomenclatura e consumo de tempo

- $O(n)$: linear
  - quando n dobra, o tempo dobra
  - em geral, uma certa quantidade de operações é realizada sobre cada um dos elementos de entrada.
  - Ex: Busca linear, Encontrar o máximo/mínimo de um vetor, Produto interno de dois vetores

- $O(n \lg n)$: log linear
  - quando n dobra, o tempo um pouco mais que dobra
  - Ex: algoritmos de ordenação eficientes

- $O(n^2)$: quadrático
  - ocorre frequentemente quando os dados são processados aos pares, com laços de repetição aninhados.
  - quando n dobra, o tempo quadriplica
  - Ex: BubbleSort, SelectionSort e InsertionSort

Nomenclatura e consumo de tempo

- $O(n)$: linear
  - quando n dobra, o tempo dobra
  - em geral, uma certa quantidade de operações é realizada sobre cada um dos elementos de entrada.
  - Ex: Busca linear, Encontrar o máximo/mínimo de um vetor, Produto interno de dois vetores

- $O(n \lg n)$: log linear
  - quando n dobra, o tempo um pouco mais que dobra
  - Ex: algoritmos de ordenação eficientes

- $O(n^2)$: quadrático
  - ocorre frequentemente quando os dados são processados aos pares, com laços de repetição aninhados.
  - quando n dobra, o tempo quadriplica
  - Ex: BubbleSort, SelectionSort e InsertionSort

Nomenclatura e consumo de tempo

- $O(n)$: linear
  - quando n dobra, o tempo dobra
  - em geral, uma certa quantidade de operações é realizada sobre cada um dos elementos de entrada.
  - Ex: Busca linear, Encontrar o máximo/mínimo de um vetor, Produto interno de dois vetores

- $O(n \lg n)$: log linear
  - quando n dobra, o tempo um pouco mais que dobra
  - Ex: algoritmos de ordenação eficientes

- $O(n^2)$: quadrático
  - ocorre frequentemente quando os dados são processados aos pares, com laços de repetição aninhados.
  - quando n dobra, o tempo quadriplica
  - Ex: BubbleSort, SelectionSort e InsertionSort

Nomenclatura e consumo de tempo

- $O(n^3)$: cúbico
  - quando n dobra, o tempo octuplica
  - Ex: multiplicação de matrizes $n \times n$

- $f(n) = O(c^n)$: complexidade exponencial
  - Típico de algoritmos que fazem busca exaustiva (força bruta) para resolver um problema.
  - Não são úteis do ponto de vista prático.
    - Quando n é 20, $O(2^n)$ é um milhão.

- $f(n) = O(n!)$: complexidade exponencial
  - Pior que $O(c^n)$
  - Não são úteis do pronto de vista prático.
  - Quando n é 20, $O(n!)$ é maior que 2 quintilhões.

Nomenclatura e consumo de tempo

- $O(n^3)$: cúbico
  - quando n dobra, o tempo octuplica
  - Ex: multiplicação de matrizes $n \times n$

- $f(n) = O(c^n)$: complexidade exponencial
  - Típico de algoritmos que fazem busca exaustiva (força bruta) para resolver um problema.
  - Não são úteis do ponto de vista prático.
    - Quando n é 20, $O(2^n)$ é um milhão.

- $f(n) = O(n!)$: complexidade exponencial
  - Pior que $O(c^n)$
  - Não são úteis do pronto de vista prático.
  - Quando n é 20, $O(n!)$ é maior que 2 quintilhões.

Nomenclatura e consumo de tempo

- $O(n^3)$: cúbico
  - quando n dobra, o tempo octuplica
  - Ex: multiplicação de matrizes $n \times n$

- $f(n) = O(c^n)$: complexidade exponencial
  - Típico de algoritmos que fazem busca exaustiva (força bruta) para resolver um problema.
  - Não são úteis do ponto de vista prático.
    - Quando n é 20, $O(2^n)$ é um milhão.

- $f(n) = O(n!)$: complexidade exponencial
  - Pior que $O(c^n)$
  - Não são úteis do pronto de vista prático.
  - Quando n é 20, $O(n!)$ é maior que 2 quintilhões.

Comparação de funções de complexidade

Tamanho	Função de custo					
n	$\lg_2 n$	n	$n \lg_2 n$	n^2	n^3	2^n
10	3	10	30	100	1000	1000
100	6	100	664	10^4	10^6	10^{30}
1000	9	1000	9965	10^6	10^9	10^{300}
10^4	13	10^4	10^5	10^8	10^{12}	10^{3000}
10^5	16	10^5	10^6	10^{10}	10^{15}	10^{30000}
10^6	19	10^6	10^7	10^{12}	10^{18}	10^{300000}

1 semana $\approx 6 \cdot 10^5$ segundos
1 ano $\approx 3 \cdot 10^7$ segundos
1 século $\approx 3 \cdot 10^9$ segundos
1 milênio $\approx 3 \cdot 10^{10}$ segundos

Um cuidado

O que significa dizer que o tempo de um algoritmo é $O(n^3)$?

- Para instâncias grandes ($n \geq n_0$)
- O tempo é menor ou igual a um múltiplo de n^3

Pode ser que o tempo do algoritmo seja $2n^2$...

- $2n^2 = O(n^3)$, mas...
- $2n^2 = O(n^2)$

Ou seja, podemos ter feito uma análise "folgada"

- achamos que o algoritmo é muito pior do que é realmente

Um cuidado

O que significa dizer que o tempo de um algoritmo é $O(n^3)$?

- Para instâncias grandes ($n \geq n_0$)
- O tempo é menor ou igual a um múltiplo de n^3

Pode ser que o tempo do algoritmo seja $2n^2$...

- $2n^2 = O(n^3)$, mas...
- $2n^2 = O(n^2)$

Ou seja, podemos ter feito uma análise "folgada"

- achamos que o algoritmo é muito pior do que é realmente

Um cuidado

O que significa dizer que o tempo de um algoritmo é $O(n^3)$?

- Para instâncias grandes ($n \geq n_0$)
- O tempo é menor ou igual a um múltiplo de n^3

Pode ser que o tempo do algoritmo seja $2n^2$...

- $2n^2 = O(n^3)$, mas...
- $2n^2 = O(n^2)$

Ou seja, podemos ter feito uma análise "folgada"

- achamos que o algoritmo é muito pior do que é realmente

Conclusão

Conclusão

- A análise de algoritmos é útil para definir o algoritmo mais eficiente em determinados problemas.
- O objetivo final não é apenas fazer códigos que funcionem, mas que sejam também eficientes.

> "Um bom algoritmo, mesmo rodando em uma máquina lenta, sempre acaba derrotando (para instâncias grandes do problema) um algoritmo pior rodando em uma máquina rápida. Sempre."
>
> — S. S. Skiena, The Algorithm Design Manual

Exercícios

Exercício

Para cada uma das afirmações abaixo, justifique formalmente (usando definições, manipulações algébricas e implicações) se for verdade ou dê um contraexemplo se for falso.

(a) $3n = O(n)$

(b) $2n^2 - n = O(n^2)$

(c) $\log 8n = O(\log 2n)$

(d) $2^{n+1} = O(2^n)$

(e) $2^n = O(2^{n/2})$

(f) $n^2 - 200n - 300 = O(n)$

(g) Se $f(n) = 17$, então $f(n) = O(1)$

(h) Se $f(n) = 3n^2 - n + 4$, então $f(n) = O(n^2)$

Exercício

Determine a complexidade de pior caso da função a seguir:

Algoritmo 3 Função F

```
Função F(int L[ ], int n)
    s ← 0
    para i ← 0 até n − 2 faça
        para j ← i + 1 até n − 1 faça
            if L[i] > L[j] then
                s ← s + 1
            fim if
        fim para
    fim para
    retorne s
fim Função
```

Notação Ω: exercício

- **Exercício:** Prove que $100 \lg n - 10n + 2n \lg n$ está em $\Omega(n \lg n)$.

Exercícios Resolvidos

Exercício

Exercício: Proponha um **limite superior** para a função $f(n) = 3n^2 + 18$ juntamente com constantes c e n_0 válidas.

Solução: Como limite superior, propomos a função $g(n) = n^2$ e como constantes válidas citamos $c = 4$ e $n_0 = 5$.

Vamos verificar essas constantes:

$$
\begin{aligned}
c \cdot g(n) &\geq f(n) \\
4n^2 &\geq 3n^2 + 18 \\
n^2 &\geq 18 \Rightarrow \{n \leq -\sqrt{18} \ \cup \ n \geq \sqrt{18}\}
\end{aligned}
$$

Como $c = 4$ e $n = 5 > 4.25 \approx \sqrt{18}$, então $3n^2 + 18 = O(n^2)$.

Exercício

Exercício: Proponha um **limite superior** para a função $f(n) = 3n^2 + 18$ juntamente com constantes c e n_0 válidas.

Solução: Como limite superior, propomos a função $g(n) = n^2$ e como constantes válidas citamos $c = 4$ e $n_0 = 5$.

Vamos verificar essas constantes:

$$\begin{aligned} c \cdot g(n) &\geq f(n) \\ 4n^2 &\geq 3n^2 + 18 \\ n^2 &\geq 18 \Rightarrow \{n \leq -\sqrt{18} \ \cup \ n \geq \sqrt{18}\} \end{aligned}$$

Como $c = 4$ e $n = 5 > 4.25 \approx \sqrt{18}$, então $3n^2 + 18 = O(n^2)$.

Exercício

Exercício: Proponha um **limite superior** para a função $f(n) = 3n^2 + 18$ juntamente com constantes c e n_0 válidas.

Solução: Como limite superior, propomos a função $g(n) = n^2$ e como constantes válidas citamos $c = 4$ e $n_0 = 5$.

Vamos verificar essas constantes:

$$\begin{aligned} c \cdot g(n) &\geq f(n) \\ 4n^2 &\geq 3n^2 + 18 \\ n^2 &\geq 18 \Rightarrow \{n \leq -\sqrt{18} \ \cup \ n \geq \sqrt{18}\} \end{aligned}$$

Como $c = 4$ e $n = 5 > 4.25 \approx \sqrt{18}$, então $3n^2 + 18 = O(n^2)$.

Exercício

Exercício: Proponha um **limite superior** para a função $f(n) = 3n^2 + 18$ juntamente com constantes c e n_0 válidas.

Solução: Como limite superior, propomos a função $g(n) = n^2$ e como constantes válidas citamos $c = 4$ e $n_0 = 5$.

Vamos verificar essas constantes:

$$\begin{aligned} c \cdot g(n) &\geq f(n) \\ 4n^2 &\geq 3n^2 + 18 \\ n^2 &\geq 18 \Rightarrow \{n \leq -\sqrt{18} \;\cup\; n \geq \sqrt{18}\} \end{aligned}$$

Como $c = 4$ e $n = 5 > 4.25 \approx \sqrt{18}$, então $3n^2 + 18 = O(n^2)$.

Exercício

Exercício: Proponha um **limite superior** para a função $f(n) = 3n^2 + 18$ juntamente com constantes c e n_0 válidas.

Solução: Como limite superior, propomos a função $g(n) = n^2$ e como constantes válidas citamos $c = 4$ e $n_0 = 5$.

Vamos verificar essas constantes:

$$\begin{aligned} c \cdot g(n) &\geq f(n) \\ 4n^2 &\geq 3n^2 + 18 \\ n^2 &\geq 18 \Rightarrow \{n \leq -\sqrt{18} \;\cup\; n \geq \sqrt{18}\} \end{aligned}$$

Como $c = 4$ e $n = 5 > 4.25 \approx \sqrt{18}$, então $3n^2 + 18 = O(n^2)$.

Exercício

Exercício: Suponha $f(n) = 2n^2 + 30n + 400$ e $g(n) = n^2$. Mostre que $f = O(g)$.

Solução: Para todo n positivo, temos:

$$
\begin{aligned}
f(n) &= 2n^2 + 30n + 400 \\
&\leq 2n^2 + 30n^2 + 400n^2 \\
&= 432n^2 \\
&= 432g(n).
\end{aligned}
$$

Resumindo, $f(n) \leq 432g(n)$ para todo $n \leq 1$. Além disso, note que $f(n)$ e $g(n)$ são assintoticamente não-negativas. Portanto, $f(n) = O(g(n))$.

Exercício

Exercício: Suponha $f(n) = 2n^2 + 30n + 400$ e $g(n) = n^2$. Mostre que $f = O(g)$.

Solução: Para todo n positivo, temos:

$$
\begin{aligned}
f(n) &= 2n^2 + 30n + 400 \\
&\leq 2n^2 + 30n^2 + 400n^2 \\
&= 432n^2 \\
&= 432g(n).
\end{aligned}
$$

Resumindo, $f(n) \leq 432g(n)$ para todo $n \leq 1$. Além disso, note que $f(n)$ e $g(n)$ são assintoticamente não-negativas. Portanto, $f(n) = O(g(n))$.

Exercício

Exercício: Suponha $f(n) = \lceil n/2 \rceil + 10$ e $g(n) = n$. Mostre que $f(n) = O(g(n))$.

Solução: De fato, temos que:

$$\begin{aligned} f(n) &= \lceil n/2 \rceil + 10 \\ &\leq n/2 + 1 + 10 \\ &= n/2 + 11 \\ &\leq 20n \text{ para todo } n \geq 1. \end{aligned}$$

Portanto, $f(n) = O(g(n))$.

Exercício

Exercício: Suponha $f(n) = \lceil n/2 \rceil + 10$ e $g(n) = n$. Mostre que $f(n) = O(g(n))$.

Solução: De fato, temos que:

$$
\begin{aligned}
f(n) &= \lceil n/2 \rceil + 10 \\
&\leq n/2 + 1 + 10 \\
&= n/2 + 11 \\
&\leq 20n \text{ para todo } n \geq 1.
\end{aligned}
$$

Portanto, $f(n) = O(g(n))$.

Exercício

Exercício: Suponha $f(n) = 5n\lg n + 8\lg^2 n - 11$ e $g(n) = n\lg n$. Mostre que $f(n) = O(g(n))$, sem usar limites.

Solução:

$$\begin{aligned} 5n\lg n + 8\lg^2 n - 11 &\leq 5n\lg n + 8\lg^2 n \\ &\leq 5n\lg n + 8n\lg n \text{ pois } \lg n < n \quad \forall n \geq 1 \\ &= 13n\lg n \end{aligned}$$

Logo, concluímos que $5n\lg n + 8\lg^2 n - 11 \leq 13n\lg n$ para todo $n \geq 1$. Portanto, fazendo $n_0 = 1$ e $c = 13$, temos que $0 \leq f(n) \leq 13g(n)$ para todo $n \geq n_0$. Assim, $f(n) = O(g(n))$.

Exercício

Exercício: Suponha $f(n) = 5n \lg n + 8 \lg^2 n - 11$ e $g(n) = n \lg n$. Mostre que $f(n) = O(g(n))$, sem usar limites.

Solução:

$$\begin{aligned} 5n \lg n + 8 \lg^2 n - 11 &\leq 5n \lg n + 8 \lg^2 n \\ &\leq 5n \lg n + 8n \lg n \text{ pois } \lg n < n \quad \forall n \geq 1 \\ &= 13n \lg n \end{aligned}$$

Logo, concluímos que $5n \lg n + 8 \lg^2 n - 11 \leq 13n \lg n$ para todo $n \geq 1$. Portanto, fazendo $n_0 = 1$ e $c = 13$, temos que $0 \leq f(n) \leq 13g(n)$ para todo $n \geq n_0$. Assim, $f(n) = O(g(n))$.

Exercícios

1. É verdade que $\log_2 n = O(\log_3 n)$? É verdade que $\log_3 n = O(\log_2 n)$?

2. Mostre que $15n = O(n \lg n)$ mas que $n \lg n \neq O(n)$
   - Essa análise é folgada, já que $15n = O(n)$

3. Mostre que $42n = O(n^2)$ mas que $n^2 \neq O(42n)$
   - Essa análise é folgada, já que $42n = O(n)$

Exercícios

1. É verdade que $\log_2 n = O(\log_3 n)$? É verdade que $\log_3 n = O(\log_2 n)$?

2. Mostre que $15n = O(n \lg n)$ mas que $n \lg n \neq O(n)$
   - Essa análise é folgada, já que $15n = O(n)$

3. Mostre que $42n = O(n^2)$ mas que $n^2 \neq O(42n)$
   - Essa análise é folgada, já que $42n = O(n)$

Exercícios

1. É verdade que $\log_2 n = O(\log_3 n)$? É verdade que $\log_3 n = O(\log_2 n)$?

2. Mostre que $15n = O(n \lg n)$ mas que $n \lg n \neq O(n)$
   - Essa análise é folgada, já que $15n = O(n)$

3. Mostre que $42n = O(n^2)$ mas que $n^2 \neq O(42n)$
   - Essa análise é folgada, já que $42n = O(n)$

Exercícios

1. É verdade que $\log_2 n = O(\log_3 n)$? É verdade que $\log_3 n = O(\log_2 n)$?

2. Mostre que $15n = O(n \lg n)$ mas que $n \lg n \neq O(n)$
   - Essa análise é folgada, já que $15n = O(n)$

3. Mostre que $42n = O(n^2)$ mas que $n^2 \neq O(42n)$
   - Essa análise é folgada, já que $42n = O(n)$

Exercícios

1. É verdade que $\log_2 n = O(\log_3 n)$? É verdade que $\log_3 n = O(\log_2 n)$?

2. Mostre que $15n = O(n \lg n)$ mas que $n \lg n \neq O(n)$
   - Essa análise é folgada, já que $15n = O(n)$

3. Mostre que $42n = O(n^2)$ mas que $n^2 \neq O(42n)$
   - Essa análise é folgada, já que $42n = O(n)$