

Princípios da Análise de Algoritmos, Indução Matemática

o exemplificar e/ou algs. de busca linear  
e de encontrar o mínimo

o algs. recursivos para exponenciação

1) Análise de pior caso: por que ser tão pessimista?

- Resultado independe da entrada e é mais simples de analisar,  
 ◦ em contraste com análises de melhor caso, caso médio e benchmark.
- Assim, análise de pior caso é apropriada para algoritmos de propósito geral.

o complexa e dependente

o empírica e dependente

o simples e muito dependente

2) Pouca atenção para constantes multiplicativas e termos de menor ordem:

$$10n^2 + 100n = O(n^2)$$

- Simplifica a análise.
- Se preocupar com constantes seria inapropriado, já que elas dependem de  
 ◦ detalhes de implementação, linguagem, compilador e arquitetura.
- Termos de menor ordem se tornam irrelevantes quando a entrada cresce.  
 ◦ Ex.:  $n^2 + 100n$  vs.  $n^2$  para  $n = 10000$ .  $\rightarrow n^2 + 100n = 100100000 \approx n^2$ .
- Comparação entre algoritmos é preditiva na maior parte dos casos.

se mantém

$n^2 = 100000000$   
mas e/ou essas simplificações

Obs: constantes são relevantes na prática. Mas, em geral, tem impacto menor que a escolha de algoritmos e estruturas de dados adequadas.

- Ex.: um bubbleSort super polido ainda é muito pior que qualquer mergeSort,  
 quando pensamos em entradas grandes.

- Constança do uso de insertion Sort como caso base do merge Sort

3) Foco no comportamento dos algoritmos para entradas grandes:

- Caso contrário força bruta resolve.
- Por isso, termos de menor ordem ficam insignificantes e
- o crescimento do tempo em função de  $n$  importa mais do que constantes.
  - Ex.:  $100 n \lg n$  vs.  $n^2$  para  $n = 1000000$ .

$$100 n \lg n = 100 \cdot 10^6 \cdot \lg 10^6 \approx 10^8 \cdot \lg 2^{20} = 2 \cdot 10^9 \quad \vee \quad n^2 = 10^{12} = 1000 \cdot 10^9$$

Neste curso estamos interessados em algoritmos rápidos, e usamos a seguinte definição matemática para o que é um algoritmo rápido:

- Um algoritmo cujo tempo de execução no pior caso
  - cresce devagar em função do tamanho da entrada.
- Por devagar queremos dizer perto de linear. *permissiva*
  - Uma definição mais geral diria "cresce de modo polinomial".
- Essa definição é vantajosa por apresentar duas características.
  - É tratável na análise e é preditiva na prática.

*Numa*  
Na próxima aula estudaremos análise assintótica,

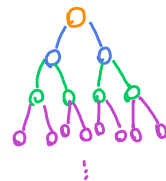
- que usaremos para formalizar vários desses princípios.

⊗ Apenas  
exemplos  
1, 2 e 6  
+ exponen-  
ciação !!!

## Indução matemática

$$\sum_{i=0}^n 2^i = 2^{n+1} - 1$$

# nós em uma  
árvore completa de  
altura n



• Exemplo 1: Mostrar que  $2^0 + 2^1 + 2^2 + \dots + 2^n = 2^{n+1} - 1$ .

Base:  $n=0 \Rightarrow \sum_{i=0}^0 2^i = 2^0 = 1 = 2 - 1 = 2^{0+1} - 1 = 2^{n+1} - 1$

H.I.: p/  $n'=n-1$  temos  $\sum_{i=0}^{n-1} 2^i = \sum_{i=0}^{n'} 2^i = 2^{n'+1} - 1 = 2^{n-1+1} - 1 = 2^n - 1$

Passo:

$$\sum_{i=0}^n 2^i = 2^n + \sum_{i=0}^{n-1} 2^i \stackrel{\text{H.I.}}{=} 2^n + 2^n - 1 = 2 \cdot 2^n - 1 = 2^{n+1} - 1$$

num raciocínio  
"recursivo", vamos  
procurar a H.I.  
dentro do que  
queremos  
provar

Na demonstração por indução queremos provar que:  $\forall n \in \mathbb{Z}^+ (P(n))$

- uma propriedade P associada a um parâmetro n, i.e.,  $P(n)$ ,
- vale para todo número natural n, i.e.,  $n = 0, 1, 2, 3, \dots$

Para tanto seguimos a seguinte estrutura:

- Base: mostramos que  $P(0)$  vale.  $P(0)$  vale
  - H.I.: supomos que  $P(n-1)$  vale.
  - Passo: mostramos que  $P(n)$  vale a partir da Hipótese de Indução (H.I.).
- $\left. \begin{array}{l} P(n-1) \Rightarrow P(n) \end{array} \right\}$

- Exemplo 2: Mostrar que a soma dos n primeiros naturais ímpares é  $n^2$ .

$$1 + 3 + 5 + 7 + \dots$$



Base:  $m=1 \Rightarrow \sum_{i=1}^m (2i-1) = 2 \cdot 1 - 1 = 1 = 1^2 = m^2$

H.I.:  $p/m' = m-1$  Temos  $\sum_{i=1}^{m-1} (2i-1) = \sum_{i=1}^{m'} (2i-1) = m'^2 = (m-1)^2$

$$\sum_{i=1}^n (2i-1) \stackrel{?}{=} m^2$$

Passo:

$$\sum_{i=1}^m (2i-1) = (2m-1) + \sum_{i=1}^{m-1} (2i-1) \stackrel{H.I.}{=} (2m-1) + (m-1)^2 = m^2 - \cancel{2m} + \cancel{1} + \cancel{2m} - \cancel{1} = m^2$$

(Pulcr?) Exemplo 3: Mostrar que para naturais x e n,  $x^n - 1$  é divisível por  $x - 1$ .  $\exists k \in \mathbb{Z} (k(x-1) = x^n - 1)$

Base:  $m=1 \Rightarrow x^m - 1 = x^1 - 1 = x - 1 = k(x-1)$   $p/ k=1$

H.I.:  $p/m' = m-1$  Temos  $\exists k \in \mathbb{Z} (k(x-1) = x^{m'} - 1 = x^{m-1} - 1)$

Passo:  $x^m - 1 = x x^{m-1} - 1 = x(k(x-1) + 1) - 1 = xk(x-1) + (x-1)$

$$\stackrel{H.I.}{=} (x-1)(\underbrace{kx+1}_{\in \mathbb{Z}}) = (x-1)k' \quad p/ k' = kx+1$$

Exemplo 4: Mostrar que  $(1+x)^n \geq 1+nx$  p/ todo natural  $n$  e real  $x$  com  $1+x > 0$ .

Base:  $n=1 \Rightarrow (1+x)^1 = (1+x)^1 = 1+1x = 1+nx$

H.I.: p/  $n'=n-1$  temos  $(1+x)^{n-1} = (1+x)^{n-1} \geq 1+n'x = 1+(n-1)x$

Passo:  $(1+x)^n = (1+x)(1+x)^{n-1} \geq (1+x)(1+(n-1)x) = 1+x + (1+x)(n-1)x = 1+x + (n-1)x + (n-1)x^2 = 1+nx + (n-1)x^2$

H.I.  $= 1+x + nx + nx^2 - x - x^2 = 1+nx + x^2(n-1) \geq 1+nx$

Exemplo 5: Mostrar que o número de regiões do plano cortado por  $n$  retas em posição geral é  $T(n) = n(n+1)/2 + 1$ .

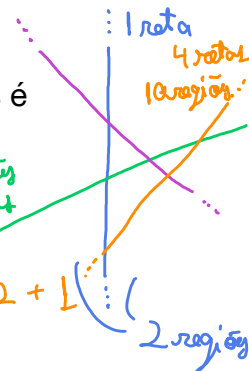
- Obs: um conjunto de retas está em posição geral se nenhum par de retas é paralelo e três ou mais retas não se cruzam num mesmo ponto.

Base:  $n=1 \Rightarrow T(1) = T(1) = 2 = 1(1+1)/2 + 1 = 2/2 + 1 = 2$

H.I.: p/  $n'=n-1$  temos  $T(n-1) = T(n') = n'(n'+1)/2 + 1 = (n-1)n/2 + 1$

Passo:  $T(n) = T(n-1) + n = (n-1)n/2 + 1 + n = \frac{n}{2}(n-1) + 1 + \frac{2n}{2}$

$= \frac{n}{2}(n-1+2) + 1 = n(n+1)/2 + 1$



- Exemplo 6: Mostrar que  $S_n = 1/2 + 1/4 + 1/8 + \dots + 1/2^n < 1$ .

$$\boxed{1/2 \quad | \quad 1/4 \quad | \quad 1/8 \quad | \quad \dots}$$

Base:  $n=1 \Rightarrow S_n = S_1 = 1/2 < 1$

H.I.:  $S_{n-1} = \sum_{i=1}^{n-1} 1/2^i < 1$  ⊗ Testar o erro no próximo

Passo: 
$$S_n = \sum_{i=1}^n 1/2^i = \frac{1}{2} \sum_{i=0}^{n-1} \frac{1}{2^i} = \frac{1}{2} \left( 1 + \sum_{i=1}^{n-1} \frac{1}{2^i} \right) < \frac{1}{2} (1 + 1) = \frac{2}{2} = 1$$

Exemplo 7: Mostrar que se  $T(n) = 4T(n/2) + cn$  então  $T(n) = 4^i T(n/2^i) + (2^i - 1)cn$ ,  $\forall i \in \mathbb{Z}^+$  e  $\forall n \in \mathbb{R}^+$

Base:  $i=1 \Rightarrow T(n) = 4T(n/2) + cn = 4^i T(n/2^i) + (2^i - 1)cn$

H.I.:  $p(i) = i-1$  Temos  $\forall n (T(n) = 4T(n/2) + cn) \Rightarrow T(n) = 4^{i-1} T(n/2^{i-1}) + (2^{i-1} - 1)cn$

Passo: 
$$\begin{aligned} T(n) &= 4T(n/2) + cn = 4 \left( 4^{i-1} T\left(\frac{n}{2^{i-1}}\right) + (2^{i-1} - 1)cn \right) + cn \\ &= 4^i T(n/2^i) + 2(2^{i-1} - 1)cn + cn \\ &= 4^i T(n/2^i) + cn(2^i - 2 + 1) = 4^i T(n/2^i) + (2^i - 1)cn \end{aligned}$$

Dados inteiros  $a$  e  $b$ , c/  $a \neq 0$  e  $b \geq 0$ , devolva  $a^b$   
 $\hookrightarrow b \in \mathbb{N}$

1ª Tentativa:

$$a^b = \underbrace{a^{b-1}}_{\text{resolva isso recursivamente (e comece no H.I.)}} \cdot a \Rightarrow$$

alg RecSimples (int  $a$ , nat  $b$ ):  
 se  $b = 0$  devolva 1  
 aux = alg RecSimples ( $a$ ,  $b-1$ )  
 devolva  $a \cdot \text{aux}$

⊗ Apresentar no final da aula 2 sobre indução, ou no começo da aula 5 sobre D&C e merge Sort. Talvez fazer o simples na 2 e o esperto na 5!!!

Provando corretude usando ind. matemática

- Base: P/  $b = 0$  temos alg RecSimples( $a, b$ ) devolve  $1 = a^0 = a^b$  ✓
- H.I.: P/  $b' = b-1$  temos alg RecSimples( $a, b'$ ) devolve  $a^{b'} = a^{b-1}$
- Passo: P/  $b$  temos alg RecSimples( $a, b$ ) devolve  $a \cdot \text{aux} \stackrel{\text{H.I.}}{=} a \cdot a^{b-1} = a^b$  ✓

Recorrência de Tempo  $T(a, b) = T(a, b-1) + c$  e  $T(a, 0) = c$

Expandindo

$$\begin{aligned} T(b) &= T(b-1) + c \\ T(b-1) &= T(b-2) + c \\ T(b-2) &= T(b-3) + c \\ &\vdots \end{aligned}$$

Substituindo

$$\begin{aligned} T(b) &= T(b-1) + c \\ &= T(b-2) + 2c \\ &= T(b-3) + 3c \\ &\vdots \end{aligned}$$

Generalizando

$$\begin{aligned} T(b) &= T(b-i) + ic \\ \text{Base da recursão} \\ T(b-i) &= T(0) = c \end{aligned}$$

$$b-i = 0 \Leftrightarrow b = i$$

Finalizando  $T(b) = T(b-i) + ic \stackrel{i=b}{=} T(0) + b \cdot c = (b+1) \cdot c = O(b) = \text{linear em } b$

↓  
 Note que bastam  $K$  bits para representar  $b \leq 2^K - 1$

Por isso, este algoritmo

$O(b) = O(2^K)$  é exponencial no tamanho de uma parte da entrada

inteiros  $a$  e  $b$ ,  $c/ a \neq 0$  e  $b \geq 0$ , devolva  $a^b$   
 $\hookrightarrow b \in \mathbb{N}$

ativa:

$= (a^{\lfloor b/2 \rfloor}) (a^{\lfloor b/2 \rfloor})$  se  $b$  for par

$= (a^{\lfloor b/2 \rfloor}) (a^{\lfloor b/2 \rfloor}) \cdot a$  se  $b$  for ímpar

Exemplos:

$\circ b=6 \quad a^6 = a^3 \cdot a^3$

$\circ b=7 \quad a^7 = a^3 \cdot a^3 \cdot a$

Intuição



D&C( $a, b$ ):

$b == 0$ : devolva 1

$= \text{algRec D\&C}(a, \lfloor b/2 \rfloor)$

$= \text{aux} * \text{aux}$

$b \% 2 == 1$ :  $\text{aux} = \text{aux} * a$

devolva  $\text{aux}$

Reconstrução de Tempo

$$T(b) = 1T(b/2) + c \quad \text{e} \quad T(0) = c$$

Expandindo / Substituindo / Generalizando

$$T(b) = T(b/2^i) + ic$$

Como a divisão de  $b$  por 2 é  
 inteira, i.e., resultado é arredondado  
 p/ baixo, na penúltima chamada recursi-  
 va ( $h-1$ ) temos:

$$b/2^{h-1} = 1 \Rightarrow 2^{h-1} = b \Rightarrow h-1 = \lg b \Rightarrow h = \lg b + 1$$

Portanto

$$T(b) = T\left(\frac{b}{2^{1+\lg b}}\right) + (1+\lg b)c = T\left(\frac{1}{2}\right) + (1+\lg b)c$$

$$= c + c + c \lg b = O(\lg b)$$

Como  $\lg b =$   
 este alg. é linear  
 bits  $k$  usado n  
 n/ representar b