Análise de Algoritmos

Ronaldo Cristiano Prati

Bloco A, sala 513-2

ronaldo.prati@ufabc.edu.br

Resolução de exercícios

- Mostre que $T(n) = 2n^2 + 10$ é $O(n^2)$
  - Temos que provar que para algum c e n_0, a definição é verdadeira. Suponha que $c = 3$ (um chute!)

$$2n^2 + 10 \leq cn^2 = 3n^2$$
$$10 \leq n^2$$
$$\sqrt{10} \leq n$$

  - Como $\sqrt{10} \approx 3.16 < 4$, a última expressão é verdadeira para $n_0 = 4$
  - Portanto, se escolhermos $c = 3$ e $n_0 = 4$, temos que $T(n) = O(n^2)$.

Resolução de exercícios

- Mostre que $T(n) = n^{1/2}$ é $O(n^{2/3})$
  - Temos que provar que para algum c e n_0, a definição é verdadeira.

$$
\begin{aligned}
n^{1/2} &\leq cn^{2/3} \\
\log n^{1/2} &\leq \log cn^{2/3} \\
1/2 \log n &\leq \log c + 2/3 \log n \\
1/2 \log n - 2/3 \log n &\leq \log c \\
-1/6 \log n &\leq \log c
\end{aligned}
$$

- Como a função $\log$ é positiva para qualquer valor maior que 1, podemos escolhar $c = 1$ e $n_0 = 1$.

Resolução de exercícios

- Mostre que $T(n) = \log_a n$ é $O(\log_b n)$
  - Temos que provar que para algum c e n_0, a definição é verdadeira.

$$\log_a n \leq c \log_b n$$
$$\frac{\log_x n}{\log_x a} \leq c \frac{\log_x n}{\log_x b}$$
$$\frac{\log_x b}{\log_x a} \leq c$$
$$\log_a b \leq c$$

- Dados quaisquer valores de a e b, sempre é possível escolher uma constante c que é maior ou igual a $\log_a b$, para qualquer n positivo.

Equação de recorrência

- Na aula passada vimos que a equação o tempo de execução do algoritmo MergeSort é dado por

$$T(n) = \begin{cases} \Theta(1), & \text{se } n = 1 \\ 2T(n/2) + \Theta(n), & \text{se } n > 1 \end{cases}$$

- Essa é uma **equação de recorrência**

Equação de recorrência

- Uma equação de recorrência nos dá uma fórmula para $T(n)$ em termos de $T(\text{algo menor que } n)$
- O desafio é: Dada uma equação de recorrência para $T(n)$, encontrar uma expressão fechada para $T(n)$ (ou seja, uma expressão que não dependa de $T(\text{algo menor que } n)$)
- Por exemplo, $T(n) = O(n \log n)$

Equação de recorrência

- Na aula passada resolvemos a recorrência iterativamente "desdobrando" a equação

$$
\begin{aligned}
T(n) &= 2T(n/2) + n (\text{assumindo que } n = 2^k) \\
T(2^k) &= 2T(2^{k-1}) + 2^k \\
&= 2 \times \underbrace{2T(2^{k-2}) + 2^1 2^{k-1}}_{T(2^{k-1})} + 2^k \\
&= 2^2 \times \underbrace{2T(2^{k-3}) + 2^2 2^{k-2}}_{T(2^{k-2})} + 2^1 2^{k-1} + 2^k \\
&\vdots \\
&= 2^k T(1) + 2^{k-1} 2^1 + ... + 2^1 2^{k-1} + 2^k \\
&= 2^k + 2^{k-1} 2^1 + ... + 2^1 2^{k-1} + 2^k
\end{aligned}
$$

Equação de recorrência

$$T(2^k) = \underbrace{2^k + 2^{k-1}2^1 + ... + 2^1 2^{k-1}}_{k \text{ termos}} + 2^k$$

$$= (k+1)2^k$$

$$= k2^k + 2^k$$

- Como $2^k = n$, podemos reescrever

$$T(n) = n \log n + n$$

$$T(n) = \Theta(n \log n)$$

Equação de recorrência

- Uma outra maneira de desdobrar a equação é:

$$
\begin{aligned}
T(n) &= 2T(n/2) + n \\
&= 2(2T(n/4) + n/2) + n = 2^2 T\left(n/2^2\right) + 2n \\
&= 2^3 T\left(n/2^3\right) + 3n \\
&\vdots \\
&= 2^i T\left(n/2^i\right) + in
\end{aligned}
$$

Equação de recorrência

- Fazendo $i = \log n$, e lembrando que $a^{\log_a b} = b$ temos que:

$$\begin{aligned} T(n) &= 2^{\log n} T\left(n/2^{\log n}\right) + n \log n \\ &= nT(1) + n \log n \\ &= n + n \log n \\ &= \Theta(n \log n) \end{aligned}$$

Método interativo

- O método interativo consiste em expandir a recorrência até chegar ao caso base
- O caso base sabemos calcular diretamente
  - Em geral, utilizamos o caso base em que $T(1) = 1$

Método interativo

- Considere $T(n) = T(n/2) + 1$ (tempo de execução da busca binária). Expandindo:

$$
\begin{aligned}
T(n) &= T(n/2) + 1 \\
&= (T((n/2)/2) + 1) + 1 = T\left(n/2^2\right) + 2 \\
&= \left(T\left(\left(n/2^2\right)/2\right) + 1\right) + 2 = T\left(n/2^3\right) + 3 \\
&\vdots \\
&= T\left(n/2^i\right) + i
\end{aligned}
$$

Método interativo

- Fazendo $i = \log n$, temos que:

$$\begin{aligned} T(n) &= T\left(n/2^i\right) + i \\ &= T\left(n/2^{\log n}\right) + \log n \\ &= T(1) + \log n \\ &= \Theta(\log n) \end{aligned}$$

Método interativo

- Considere $T(n) = 3T(n/2) + n$ (tempo de execução do algoritmo de Karatsuba). Expandindo:

$$
\begin{aligned}
T(n) &= 3T(n/2) + n \\
&= 3(3T(n/4) + n/2) + n = 3^2 T\left(n/2^2\right) + 2n \\
&= 3^3 T\left(n/2^3\right) + 3n \\
&\vdots \\
&= 3^i T\left(n/2^i\right) + in
\end{aligned}
$$

Método interativo

- Fazendo $i = \log n$, e lembrando que $a^{\log_c b} = b^{\log_c a}$ temos que:

$$\begin{aligned} T(n) &= 3^{\log n} T\left(n/2^{\log n}\right) + n \log n \\ &= 3^{\log n} T(1) + n \log n \\ &= 3^{\log n} + n \log n \\ &= n^{\log 3} + n \log n \\ &= \Theta(n^{\log 3}) \end{aligned}$$

Método interativo

- Considere $T(n) = 2T(n/3) + 1$. Expandindo:

$$
\begin{aligned}
T(n) &= 2T(n/3) + 1 \\
&= 2\left(2T\left(n/3^2\right) + 1\right) + 1 = 2^2 T\left(n/3^2\right) + (1+2) \\
&= 2^3 T\left(n/3^3\right) + \left(1 + 2 + 2^2\right) \\
&\vdots \\
&= 2^i T\left(n/3^i\right) + \sum_{j=0}^{i-1} 2^j \\
&= 2^i T\left(n/3^i\right) + 2^i - 1
\end{aligned}
$$

Método interativo

- Fazendo $i = \log_3 n$, temos que:

$$\begin{aligned} T(n) &= 2T(1) \times 2^{\log_3 n} - 1 \\ &= 2\left(2^{\log n}\right)^{1/\log 3} - 1 \\ &= 2n^{1/\log 3} - 1 \\ &= \Theta\left(n^{1/\log 3}\right) \end{aligned}$$

Método interativo

- Nem sempre o método iterativo para resolução de recorrências funciona bem.
- Quando o tempo de execução de um algoritmo é descrito por uma recorrência nao tao balanceada como a dos exemplos dados, pode ser dificil aplicar esse metodo.
- Outro ponto fraco é que rapidamente os calculos podem ficar complicados.

Método de substituição

- Esse metodo consiste simplesmente em provar por indução matemática que uma recorrência $T(n)$ é limitada (inferiormente e/ou superiormente) por alguma função $g(n)$.
- Um ponto importante é que, como é uma prova por indução, é necessario que se saiba qual é a funçao $g(n)$ de antemão.

Método de substituição

- Vamos provar por indução que $T(n) = 2T(n/2) + n$ é $O(n^2)$
- Temos que provar que existem constantes c e n_0 tal que se $n > n_0$, então $T(n) \leq cn^2$
  - Caso base: $T(1) = 1$
  - Hipótese de indução: $T(m) \leq m^2$ para $1 \leq m < n$
  - Passo de indução: $T(n) \leq n^2$

Método de substituição

- Assumindo que a hipótese é verdadeira e tomando $m = n/2$, temos que:

$$\begin{aligned} T(n) &= 2T(n/2) + n \\ &\leq 2\left(\frac{n^2}{2^2}\right) + n \\ &= \left(n^2/2\right) + n \\ &\leq n^2 \end{aligned}$$

- em que a última desigualdade vale sempre que $n \geq 2$, que é o caso.
- Portanto, mostramos por indução que $T(n) \leq cn^2$ para $c \geq 1$ e $n \geq n_0 = 1$, de onde concluímos que $T(n)$ é $O(n^2)$

Método de substituição

- Provamos por substituição que $T(n) = 2T(n/2) + n$ é $O(n^2)$
- Entretanto, utilizando o método iterativo, sabemos que $T(n) = 2T(n/2) + n$ é $O(n \log n)$
- Como no método por substituição temos que saber de antemão quem é a $g(n)$, eventualmente podemos não utilizar a $g(n)$ com menor crescimento assintótico

Método de substituição

- Vamos provar por indução que $T(n) = 2T(n/2) + n$ é $O(n \log n)$
- Temos que provar que existem constantes c e n_0 tal que se $n > n_0$, então $T(n) \leq cn \log n$
  - Caso base: $T(1) = 1$. Entretanto, o se usarmos $n = 1$ como caso base temos que $1 < 0$ (que não é válido). Mas se usarmos $n = 2$ como base da indução temos que $T(2) = 2T(1) + 2 = 4 \leq c \cdot 2 \log 2$ para $c > 2$
  - Hipótese de indução: $T(m) \leq cm \log m$ para $2 \leq m < n$
  - Passo de indução: $T(n) \leq n \log n$

Método de substituição

- Assumindo que a hipótese é verdadeira e tomando $m = n/2$, temos que:

$$\begin{aligned} T(n) &= 2T(n/2) + n \\ &\leq 2(c(n/2)\log(n/2)) + n \\ &= cn\log n - cn + n \\ &\leq cn\log n, \text{ para } c \geq 1 \end{aligned}$$

- Portanto temos que $T(n) \leq cn\log n$ para $c \geq 2$ e $n \geq n_0 = 2$, e temos que $T(n) = O(n\log n)$

Método de substituição

- Uma outra possibilidade é mostrar que $T(n) = n \log n + n$. Para isso, temos que provar que existem constantes c e n_0 tal que se $n > n_0$, então $T(n) \leq cn \log n + n$
  - Caso base: $T(1) = 1$. Nesse caso temos que $1 \leq 1 \log 1 + 1$
  - Hipótese de indução: $T(m) \leq cm \log m + m$ para $1 \leq m < n$
  - Passo de indução: $T(n) \leq n \log n$

Método de substituição

- Assumindo que a hipótese é verdadeira e tomando $m = n/2$, temos que:

$$\begin{aligned} T(n) &= 2T(n/2) + n \\ &\leq 2((n/2)\log(n/2) + n/2) + n \\ &= n\log(n/2) + 2n \\ &= n\log n - n + 2n \\ &= n\log n + n \end{aligned}$$

- Portanto temos que $T(n) = O(n \log n)$.

Método de substituição

- Para garantir que não consiguimos diminiru a ordem de grandeza de $T(n)$, temos que provar que $T(n) = \Omega(n \log n)$. Para isso, temos que provar que existem constantes c_2 e n_0 tal que se $n > n_0$, então $c_2 n \log n \leq T(n)$
  - Caso base: $T(1) = 1$. Nesse caso temos que $1 \geq 1 \log 1$
  - Hipótese de indução: $T(m) \geq m \log m$ para $1 \leq m < n, c_2 = 1$
  - Passo de indução: $T(n) \geq n \log n$

Método de substituição

- Assumindo que a hipótese é verdadeira e tomando $m = n/2$, temos que:

$$\begin{aligned} T(n) &= 2T(n/2) + n \\ &\geq 2((n/2)\log(n/2)) + n \\ &= n \log n \end{aligned}$$

- Portanto temos que $T(n) = \Omega(n \log n)$

Método de substituição

(exemplo)

- Prove que $T(n) = 4T(n/2) + n^3$ é $\Theta(n^3)$.
- Primeiramente vamos mostrar que $T(n) = O(n^3)$, ou seja $T(n) \leq cn^3$ para alguma constate c e $n > n_0$:
  - Caso base: $T(1) = 1$. Nesse caso temos que $1 \leq c1^3, \forall c \geq 1$
  - Hipótese de indução: $T(m) \geq cm^3$ para $2 \leq m < n$
  - Passo de indução: $T(n) \leq cn^3$

Método de substituição

(exemplo)

- Assumindo que a hipótese é verdadeira e tomando $m = n/2$, temos que:

$$\begin{aligned} T(n) &= 4T(n/2) + n^3 \\ &\leq \frac{4cn^3}{8} + n^3 \\ &\leq cn^3 \end{aligned}$$

- que é verdadeira sempre que $c \geq 2$. Portando escolher $c = 2$ (ou qualquer valor maior), provamos por indução que $T(n) = O(n^3)$

Método de substituição

(exemplo)

- Para mostrar que $T(n) = \Omega(n^3)$, ou seja $T(n) \geq c_2 n^3$ para alguma constate c_2 e $n > n_0$:
  - Caso base: $T(1) = 1$. Nesse caso temos que $1 \geq c_2 1^3, \forall c_2 \leq 1$
  - Hipótese de indução: $T(m) \geq c_2 m^3$ para $2 \leq m < n$
  - Passo de indução: $T(n) \geq c_2 n^3$

Método de substituição

(exemplo)

- Assumindo que a hipótese é verdadeira e tomando $m = n/2$, temos que:

$$\begin{aligned} T(n) &= 4T(n/2) + n^3 \\ &\geq \frac{4c_2 n^3}{8} + n^3 \\ &\geq c_2 n^3 \end{aligned}$$

  - que é verdadeira sempre que $c_2 \leq 2$. Portando escolher $c_2 = 1$, provamos por indução que $T(n) = \Omega(n^3)$