Algoritmos e Estrutura de Dados

Aula 04 – Recorrência

Prof. Tiago A. E. Ferreira

Esta Aula...

- Nesta aula veremos três métodos para resolver recorrência:
  - Método da substituição
    - É suposto um limite hipotético e depois usa-se a indução matemática para se provar a suposição
  - Método de Árvore de Recursão
    - Converte-se a recorrência em uma árvore, onde esta é utilizada para o cálculo dos custos
  - Método Mestre
    - Fornece limites para recorrências da forma $T(n)=aT(n/b)+f(n)$, onde $a, b>0$ e $f(n)$ é uma função dada.

Método da Substituição

- Engloba duas etapas:
  - Pressupor a forma da solução
  - Usar a indução matemática para encontrar as constantes e mostrar que a solução funciona
- Este método pode ser utilizado para estabelecer limites superiores ou inferiores sobre uma recorrência

Exemplo

- Determinar um limite superior para a recorrência:
  - $T(n)=2T(\lfloor n/2 \rfloor)+n$
    - Supõem-se $T(n)=O(n \lg n)$
    - Quer-se provar que $T(n) \leq cn \lg n$, com $c> 0$
    - Começa-se supondo que este limite permanece válido para $\lfloor n/2 \rfloor$, ou seja, $T(\lfloor n/2 \rfloor) \leq c \lfloor n/2 \rfloor \lg(\lfloor n/2 \rfloor)$
    - Na recorrência,

$$
\begin{aligned}
T(n) &\leq 2(c \lfloor n/2 \rfloor \lg(\lfloor n/2 \rfloor)) + n \\
&\leq cn \lg(n/2) + n \\
&= cn \lg n - cn \lg 2 + n \\
&= cn \lg n - cn + n \\
&\leq cn \lg n ,
\end{aligned}
$$

onde o último passo é válido desde que $c \geq 1$.

Exemplo (cont.)

- Ainda falta mostrar a validade em valores limites!
  - Vamos supor que T(1) = 1, então para n=1
    - O limite tem que ser válido: $T(n) \leq cn \lg n$, mas para n=1, $T(1) \leq c\ 1 \lg 1 = 0$, assim o caso básico deixa de ser válido!
    - Porém, devido a notação assintótica, $T(n) \leq cn \lg n$ deve ser válido para $n \geq n_0$
    - Fazendo $n_0 = 2$, os casos básicos são verdadeiros para um $c \geq 2$

Cuidado! Armadilhas!

- É fácil errar utilizando a notação assintótica!
- Ex.:
  - Dada a recorrência $T(n)=2T(\lfloor n/2 \rfloor)+n$, considere $T(n) \leq cn$, então:

$$
\begin{aligned}
T(n) &\leq 2(c\lfloor n/2 \rfloor + n \\
&\leq cn + n \\
&= O(n) , \qquad\qquad \Leftarrow \textit{errado!!}
\end{aligned}
$$

  - O erro é que não se prova a forma exata da hipótese indutiva, $T(n) \leq cn$.

Método da Árvore de Recursão

- Uma grande dificuldade do método de substituição é encontrar uma boa suposição
- Traçar uma árvore de recursão pode ser uma forma de se encontrar uma boa suposição
  - Em uma árvore de recursão, tem-se:
    - Nó – Representa o custo de um único subproblema
    - Soma-se os custos por nível da árvore
    - Soma-se os custos de todos os níveis para se obter uma estimativa do custo total
  - Estas árvores de recursão são muito úteis para algoritmos do tipo dividir e conquistar

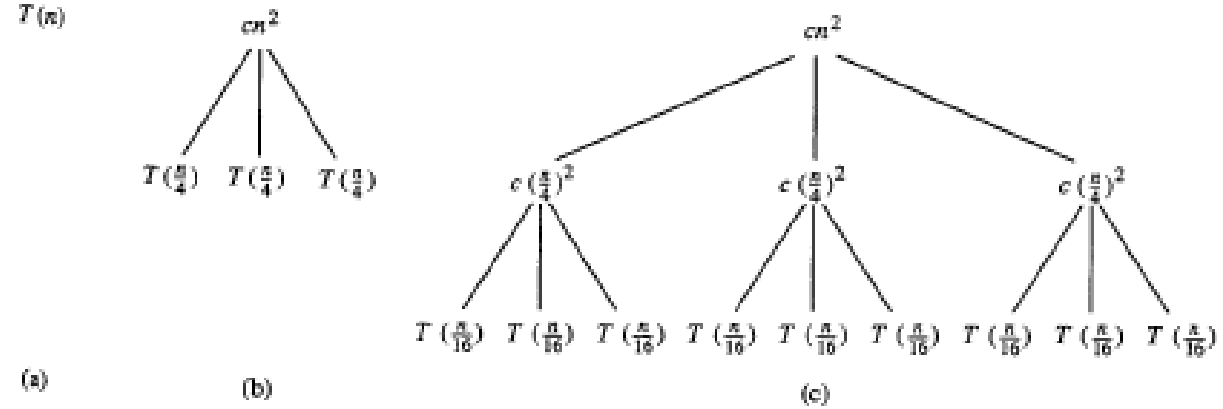
Método da Árvore de Recursão

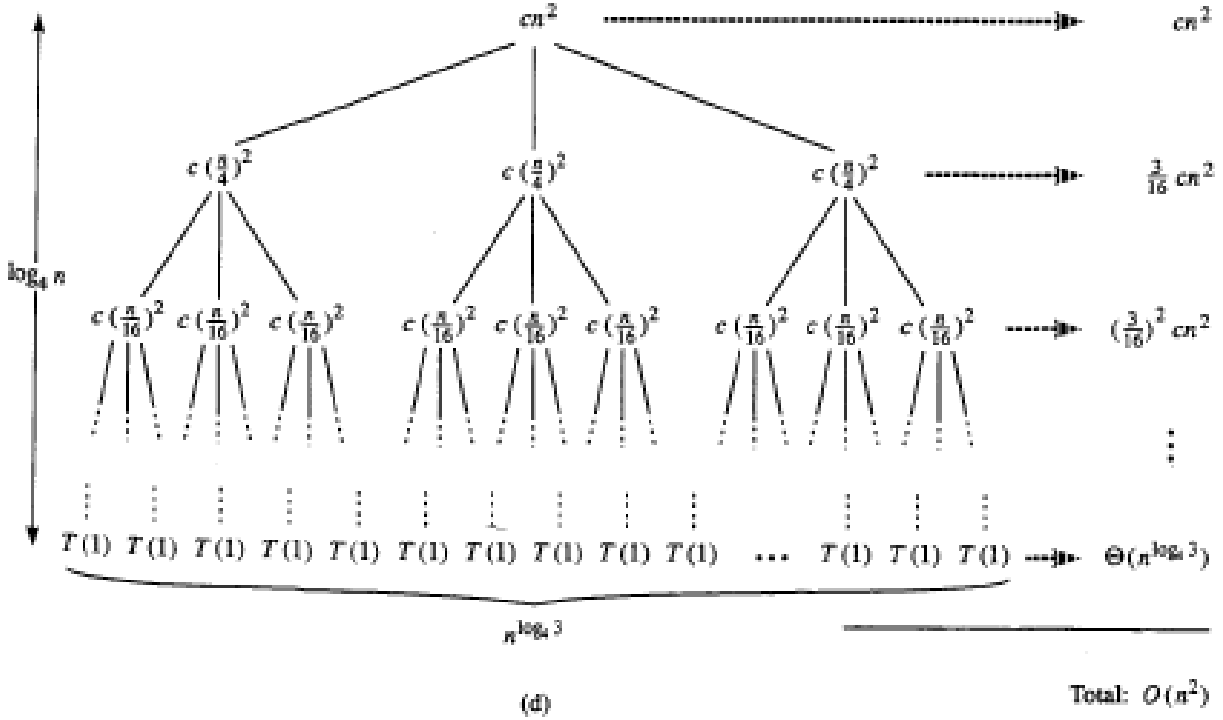
- Dependendo da forma com que se admite ou não uma certa“sujeira” nos cálculos dos custos, uma árvore de recursão tanto pode ser utilizada para:
  - A geração de uma estimativa de custo, para a criação de uma boa suposição
  - Ou, para a determinação assintótica do custo do algoritmo

Gerando uma boa suposição...

- Dada a recursão: $T(n)=3T(\lfloor n/4 \rfloor)+\Theta(n^2)$
  - Deseja-se encontrar um limite superior!
  - "Sujeira" – tolera-se o fato de que pisos e tetos são normalmente insatisfatórios para resolver recorrências
  - Cria-se uma árvore de recursão para $T(n)=3T(n/4)+cn^2$, onde $c>0$
    - Por conveniência, é suposto que *n* é uma potência de 4 (outra "sujeira")

Gerando uma boa suposição...





Analisando a Árvore

- O tamanho de um subproblema na profundidade *i* é $n/4^i$
  - *Onde* $i=\log_4(n)$
  - Logo, a árvore tem $\log_4(n)+1$ níveis $(0,1,2,3,\ldots,\log_4 n)$
- O custo de cada nível:
  - Cada nível tem 3 vezes mais nós que o nível anterior
    - Nível *i*, número de número de nós é 3^i
  - Como os tamanhos dos subproblemas são reduzidos por um fator 4 a cada nível, cada nó no nível *i* tem custo $c(n/4^i)^2$
  - Assim, o custo total para um nível *i* será $3^i \cdot c(n/4^i)^2=(3/16)^i cn^2$

Analisando a Árvore

- O custo do último nível:
  - O último nível, profundidade $\log_4 n$, tem uma quantidade de nós igual a:

$$3^{\log_4 n} = n^{\log_4 3}$$

    - Cada qual contribuindo com o custo T(1), gerando um custo total para este nível de:

$$n^{\log_4 3} T(1) \rightarrow O\left(n^{\log_4 3}\right)$$

- Custo Total:

$$T(n) = cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \cdots + \left(\frac{3}{16}\right)^{\log_4 n - 1} cn^2 + \Theta(n^{\log_4 3})$$

$$= \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3})$$

$$= \frac{(3/16)^{\log_4 n} - 1}{(3/16) - 1} cn^2 + \Theta(n^{\log_4 3})$$

Análise a Árvore

- Fazendo uso de algumas aproximações, o custo total pode ser expresso como:

$$T(n) = \sum_{i=0}^{\log_4 n-1} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3})$$

$$< \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3})$$

$$= \frac{1}{1-(3/16)} cn^2 + \Theta(n^{\log_4 3})$$

$$= \frac{16}{13} cn^2 + \Theta(n^{\log_4 3})$$

$$= O(n^2) .$$

Analisando suposição

- O próximo passo é verificar, com o método da substituição, se a suposição é boa, isto é, se $T(n)=O(n^2)$ é um limite superior para a recorrência $T(n)=3T(\lfloor n/4 \rfloor)+\Theta(n^2)$
  - Quer-se mostrar que $T(n) \leq dn^2$, $d>0$. Usando a mesma cte $c>0$,

$$
\begin{aligned}
T(n) &\leq 3T(\lfloor n/4 \rfloor) + cn^2 \\
&\leq 3d \lfloor n/4 \rfloor^2 + cn^2 \\
&\leq 3d\,(n/4)^2 + cn^2 \\
&= \frac{3}{16} dn^2 + cn^2 \\
&\leq dn^2 ,
\end{aligned}
$$

onde a última etapa é válida desde que $d \geq (16/13)c$.

Método Mestre

- O método mestre fornece uma "receita" para a resolução de recorrências na forma $T(n)=aT(n/b)+f(n)$,
  - onde $a>=1$ e $b>1$ (constantes) e $f(n)$ é uma função assintoticamente positiva
  - Esta recorrência descreve um problema de tamanho n dividido em a subproblemas de tamanho n/b
  - O custo em tempo para cada um dos subproblemas *é* $T(n/b)$ e o custo de dividir e combinar está na função $f(n)$.

Teorema Mestre

Teorema 4.1 (Teorema mestre)
Sejam $a \geq 1$ e $b > 1$ constantes, seja $f(n)$ uma função e seja $T(n)$ definida sobre os inteiros não negativos pela recorrência

$$T(n) = aT(n/b) + f(n) ,$$

onde interpretamos n/b com o significado de $\lfloor n/b \rfloor$ ou $\lceil n/b \rceil$. Então, $T(n)$ pode ser limitado assintoticamente como a seguir.

1. Se $f(n) = O(n^{\log_b a - \epsilon})$ para alguma constante $\epsilon > 0$, então $T(n) = \Theta(n^{\log_b a})$.
2. Se $f(n) = \Theta(n^{\log_b a})$, então $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. Se $f(n) = \Omega(n^{\log_b a + \epsilon})$ para alguma constante $\epsilon > 0$, e se $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e para todo n suficientemente grande, então $T(n) = \Theta(f(n))$. ■

Usando o Teorema Mestre

- **Primeiro:** Verificar a que caso o problema se aplica ao teorema (1,2 ou 3)
- **Segundo:** Anotar a resposta
- Ex.:
  - **T(n)=9T(n/3)+n**
    - a=9, b=3, f(n)=n, e $n^{\log_b a} = n^{\log_3 9} = \Theta\left(n^2\right)$
    - Como, $f(n) = O\left(n^{\log_3 9 - \varepsilon}\right), \varepsilon = 1$
    - Estamos no caso 1 do teorema: $T(n)=\Theta(n^2)$

Exemplo do uso do teorema mestre

- Considere, $T(n) = T(2n/3) + 1,$
  - a=1, b=3/2, f(n)=1 e $n^{\log_b a} = n^{\log_{3/2} 1} = n^0 = 1$
  - Assim, aplica-se o caso 2, pois $f(n) = \Theta(n^{\log_b a}) = \Theta(1)$
  - E a solução da recorrência é $T(n) = \Theta(\lg n)$

- Considere agora, $T(n) = 3T(n/4) + n \lg n$
  - a=3, b=4, $f(n) = n \lg n$ e $n^{\log_b a} = n^{\log_4 3} = O(n^{0,793})$
  - Como $f(n) = \Omega(n^{\log_4 3+\epsilon}), \epsilon \approx 0{,}2$
  - Aplica-se o caso 3, que para n suficientemente grande

    $$af(n/b) = 3(n/4)\lg(n/4) \leq (3/4)n \lg n = cf(n) \text{ para } c = 3/4$$

  - E a solução para a recorrência é $T(n) = \Theta(n \lg n)$

Contra-Exemplo

- Considere a recorrência,

$$T(n) = 2T(n/2) + n \lg n.$$

- Para este caso não é possível se aplicar o teorema mestre!
  - Observe que

$$f(n)/n^{\log_b a} = (n \lg n)/n = \lg n$$

  - Logo, f(n) é assintoticamente menor que n^{ϵ} para qualquer ϵ positiva.