Recorrências

Profa. Juliana Kaizer Vizzotto

Projeto e Análise de Algoritmos

Roteiro

- Questões Técnicas
- Método da Substituição
- Método da Árvore de Recursão
- Método *Master*

Questões Técnicas

- Na prática, negligenciamos certos detalhes técnicos quando resolvemos recorrências.
- Um bom exemplo de detalhe que é frequentemente considerado é que os argumentos das funções são números inteiros.
- Normalmente, o tempo de execução $T(n)$ de um algoritmo é somente definido quando n é um número inteiro, já que para a maioria dos algoritmos, o tamanho da entrada é sempre um inteiro.

Questões Técnicas

- Por exemplo, a recorrência descrita no pior caso do MERGE-SORT seria realmente:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + \Theta(n) & \text{if } n > 1. \end{cases}$$

- Condições de borda representam outra classe de detalhe que nós tipicamente ignoramos.
- Como o tempo de execução de um algoritmo é em uma entrada de tamanho constante é uma constante, as recorrências geralmente tem $T(n) = \Theta(1)$ para n suficientemente pequenos.

Questões Técnicas

- Consequentemente, por conveniência, geralmente omitimos as condições de fronteira nas recorrências e assumimos que $T(n)$ é constante para n pequenos.
- Por exemplo, normalmente definimos recorrências como:

$$T(n) = 2T(n/2) + \Theta(n),$$

  sem dar explicitamente os valores para n pequenos.
- A razão para isto é que embora trocando o valor de $T(1)$ troca a solução da recorrência, a solução tipicamente não troca por mais do que um fator constante, assim a ordem de crescimento é a mesma.

Questões Técnicas

- Por exemplo, a recorrência descrita no pior caso do MERGE-SORT seria realmente:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + \Theta(n) & \text{if } n > 1. \end{cases}$$

- Condições de borda representam outra classe de detalhe que nós tipicamente ignoramos.
- Como o tempo de execução de um algoritmo é em uma entrada de tamanho constante é uma constante, as recorrências geralmente tem $T(n) = \Theta(1)$ para n suficientemente pequenos.

Questões Técnicas

- Quando definimos e resolvemos recorrências, frequentemente omitidos os *pisos*, *tetos*, e *condições de fronteira*.
- Primeiro esquecemos esses detalhes e depois determinamos se eles importam ou não.
- Eles, geralmente, não importam, mas é interessante saber quando eles importam.

Método da Substituição

- O método da substituição para resolver recorrências envolve dois passos:
  1. Descubra a forma da solução.
  2. Use indução matemática para encontrar as constantes e mostre que a solução vale!
- Esse nome vem da substituição da resposta descoberta pela função quando a hipótese indutiva é aplicada em valores pequenos.
- Este método é poderoso, mas eles obviamente só pode ser aplicado nos casos em que é fácil descobrir a forma da resposta.

Método da Substituição

- O método da substituição pode ser usado para estabelecer ou *upper bounds* ou *lower bounds* em uma recorrência.
- Como um exemplo, vamos determinar um *upper bound* para a recorrência:

$$T(n) = 2T(\lfloor n/2 \rfloor) + n,$$

- Acreditamos que a solução é $T(n) = O(nlgn)$.
- Nosso método para provar que $T(n) < cnlgn$, para uma escolha apropriada de constante $c > 0$.
- Começamos assumindo (pela hipótese indutiva) que isto vale para $\lfloor n/2 \rfloor$, i.e., que $T(\lfloor n/2 \rfloor) \leq c\lfloor n/2 \rfloor lg(\lfloor n/2 \rfloor)$.

Método da Substituição

- Substituindo na recorrência, temos:

$$\begin{aligned} T(n) &\leq 2(c\lfloor n/2 \rfloor lg(\lfloor n/2 \rfloor)) + n \\ &\leq cnlg(n/2) + n \\ &= cnlgn - cnlg2 + n \\ &= cnlgn - cn + n \\ &\leq cnlgn, \end{aligned}$$

- Onde o último passo vale para $c \geq 1$.

Método da Substituição

- A indução matemática requer que mostremos que nossa solução vale para as condições de borda/fronteira.
- Tipicamente fazemos isso mostrando que as condições de borda são os casos básicos para a prova indutiva.
- Para a recorrência acima, precisamos mostrar que podemos escolher a constante c grande o suficiente de maneira que $T(n) = cnlgn$ vale para as condições de fronteira também.
- Isto pode algumas vezes levar a problemas.
- Vamos assumir que $T(1) = 1$ é a única condição de borda da recorrência.
- Então, para $n = 1$, a borda $T(n) = cnlgn$ produz $T(1) = c1lg1 = 0$, a qual é diferente de $T(1) = 1$.
- Consequentemente, o caso base da nossa prova indutiva falha! :(

Método da Substituição

- Esta dificuldade na prova da hipótese indutiva para uma condição de borda específica pode ser facilmente tratada.
- Por exemplo, na recorrência:

$$T(n) = 2T(\lfloor n/2 \rfloor) + n,$$

- temos a vantagem que a notação assintótica somente requer que provemos $T(n) = cnlgn$ para $n \geq n_0$, tal que n_0 é uma constante que escolhemos.
- Assim, a idéia é remover a condição de borda difícil $T(1) = 1$ na prova indutiva.
- Observe que para $n > 3$, a recorrência não depende diretamente de $T(1)$.

Método da Substituição

- Então, podemos substituir $T(1)$ por $T(2)$ e $T(3)$ como casos básicos na prova indutiva.
- Então, substituímos $T(1)$ por $T(2)$ e $T(3)$ como caso básico na prova indutiva, tal que $n_0 = 2$.
- Note que fizemos uma distinção entre o caso básico da recorrência $(n = 1)$ e os casos básicos da prova indutiva $(n = 2 \ e \ n = 3)$.
- Podemos derivar da recorrência que $T(2) = 4$ e que $T(3) = 5$.
- A prova indutiva que $T(n) \leq cnlgn$ para alguma constante $c \geq 1$ pode agora ficar completa escolhendo algum c grande o sufuciente de maneira que $T(2) \leq c_2 lg2$ e $T(3) \leq c_3 lg3$.
- Assim, qualquer escolha de $c \geq 2$ é suficiente para os casos básicos, $n = 2$ e $n = 3$,

Método da Substituição

- Para a maioria das recorrências que vamos trabalhar, é fácil estender as condições de borda to fazer provas indutivas

Fazendo uma Boa Descoberta

- Infelizmente, não existe um modo geral para descobir soluções para recorrências.
- Descobrir uma solução, requer experiência e, ocasionalmente, criatividade.
- Entretanto, felizmente, existem algumas heurísticas que ajudam você a se tornar um *bom descobridor* :)
- Você também pode usar o desenho da árvore de recursão.

Fazendo uma Boa Descoberta

- Se a recorrência é similar a alguma que você já trabalhou antes, então é possível descobrir uma solução similar.
- Por exemplo:

$$T(n) = 2T(\lfloor n/2 \rfloor + 17) + n,$$

  a qual parece difícil pois adicionamos o número 17 no argumento de T.
- Intuitivamente, entretanto, esse termo adicional não pode afetar substancialmente a solução da recorrência.
- **Quando n é grande, a diferença entre $T(\lfloor n/2 \rfloor)$ e $T(\lfloor n/2 \rfloor + 17$ não é muito grande.**
- Consequentemente, podemos descobrir que $T(n) = O(nlgn)$.
- **Exercício: verifique usando o método da substituição que isso vale**

Fazendo uma Boa Descoberta

- **Solução**: mostrando que:
  $T(n) = 2T(\lfloor n/2 \rfloor + 17) + n = O(nlgn)$.
- **Casos básicos**: Considere $n = 34$.
- $T(34) = 2T(17 + 17) + 34 = 2T(34) + 34$. Então
  $T(34) = 2T(34) + 34$, o que nos dá $T(34) = -34$.
- Assim, vamos assumir como casos básicos que $T(n) = 1$ para $0 < n < 35$.
- Precisamos mostrar que, para algum $n_0 > 0$ e alguma constante $c > 0$:

  $$T(n) \leq cnlgn$$

  para todo $n > n_0$. Seja $n_0 = 35$.

Fazendo uma Boa Descoberta

- Assumimos a hipótese indutiva que isto vale para $\lfloor n/2 \rfloor$, i.e., que $T(\lfloor n/2 \rfloor) \leq c(\lfloor n/2 \rfloor) lg(\lfloor n/2 \rfloor)$.
- Substituindo na recorrência, temos:

$$\begin{aligned} T(n) &\leq 2T[(\lfloor n/2 \rfloor) + (17)] + n \\ &\leq 2c[(\lfloor n/2 \rfloor) lg(\lfloor n/2 \rfloor)] + 2T(17) + n \\ &\leq 2c(n/2) lg(n/2) + 2T(17) + n \\ &= cnlgn - cn + 2T(17) + n \\ &= cnlgn - cn + 38 + n \end{aligned}$$

- Temos que achar c e n tal que

$$-cn + 38 + n \leq 0$$

- Isso vale para $c = 2$ e $n \geq 38$

Fazendo uma Boa Descoberta

- Cálculo de $T(17)$

$$\begin{aligned} T(17) &= 2 * T(34) + 17 \\ &= 19 \end{aligned}$$

Fazendo uma Boa Descoberta

- Algumas vezes quando você pode descobrir corretamente a função assintótica de uma recorrência, parece ocorrer que a indução não funciona.
- Geralmente, o problema é que a hipótese indutiva não é forte o suficiente para fazer uma prova detalhada.
- Quando você encontra esse problema, uma ideia é revisar a descoberta subtraindo um termo de baixa ordem.
- Considere a recorrência:

$$T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1$$