MAC 5711 - Análise de Algoritmos

Departamento de Ciência da Computação

Segundo semestre de 2005

Gabarito parcial da lista 2

---

1. Resolva as recorrências abaixo:

   (e) $T(n) = T(\lfloor n/3 \rfloor) + T(\lfloor 2n/3 \rfloor) + \Theta(n)$

   Resolução: A recorrência simplificada correspondente a esta é

$$T(n) \quad = \quad \begin{cases} 1 & \text{se } n = 0 \text{ ou } n = 1 \\ T(\lfloor n/3 \rfloor) + T(\lfloor 2n/3 \rfloor) + n & \text{se } n \geq 2. \end{cases}$$

Expandindo a árvore de recursão desta recorrência, percebemos que ela é $\Theta(n \lg n)$. Na verdade, obtemos da expansão até um "chute" de solução da recorrência. Algo do tipo $T(n) \leq n \log_{3/2} n$. Porém, é fácil ver que isso não vale para $n = 0$ ou $n = 1$, ou mesmo para $n = 2$. Para não termos que lidar com termos de ordem menor, vamos abrir mão de incluir na prova por indução os casos em que $n = 0$ ou $n = 1$. Vamos tentar fazer a prova por indução a partir de $n = 2$. Porém, como para $n = 2$, a fórmula acima ainda não vale, temos que ajustá-la um pouco. Depois de experimentar algumas coisas, chegamos por exemplo que é possível mostrar que

$$T(n) \leq \tfrac{3}{2} n \log_{3/2} n \quad \text{para todo } n \geq 2.$$

Prova: Por indução em n.

Base: $2 \leq n \leq 5$.

Exercício: Calcule $T(n)$ para $n = 2, 3, 4, 5$ e, com o auxílio de uma calculadora, verifique que $T(n) \leq \frac{3}{2} n \log_{3/2} n$ nestes casos.

Passo: $n \geq 6$.

Neste caso, $T(n) = T(\lfloor n/3 \rfloor) + T(\lfloor 2n/3 \rfloor) + n$, com $\lfloor n/3 \rfloor$ e $\lfloor 2n/3 \rfloor$ ambos maiores ou iguais a 2 e menores que n. Podemos então aplicar indução e obtemos que

$$\begin{aligned}
T(n) &= T(\lfloor n/3 \rfloor) + T(\lfloor 2n/3 \rfloor) + n \\
&\leq \frac{3}{2} \lfloor n/3 \rfloor \log_{3/2} \lfloor n/3 \rfloor + \frac{3}{2} \lfloor 2n/3 \rfloor \log_{3/2} \lfloor 2n/3 \rfloor + n \quad \text{por indução} \\
&\leq \frac{n}{2} \log_{3/2} \frac{n}{3} + n \log_{3/2} \frac{2n}{3} + n \\
&= \frac{n}{2} (\log_{3/2} n - \log_{3/2} 3) + n(\log_{3/2} n - \log_{3/2} 3/2) + n \\
&= \frac{3}{2} n \log_{3/2} n - \frac{n}{2} \log_{3/2} 3 - n + n \\
&= \frac{3}{2} n \log_{3/2} n - \frac{n}{2} \log_{3/2} 3 \\
&\leq \frac{3}{2} n \log_{3/2} n.
\end{aligned}$$

Isso mostra que $T(n) = O(n \lg n)$. Para concluir que $T(n) = \Theta(n \lg n)$, é preciso mostrar também que $T(n) = \Omega(n \lg n)$.

3. Escreva um algoritmo eficiente que busca um valor x em uma matriz $A_{n\times n}$ cujas linhas e colunas estão ordenadas em ordem não-decrescente. Seu algoritmo deve consumir tempo $O(n)$.

   Comentário: Seu algoritmo consumia tempo $O(n)$? Se consumia mais, procure bolar um que consuma tempo $O(n)$.

7. Considere a recorrência

$$T(n) \quad = \quad \begin{cases} 1 & \text{se } n = 0 \text{ ou } n = 1 \\ \min_{0\le k\le n-1}\{T(k) + T(n-1-k)\} + n & \text{se } n \ge 2. \end{cases}$$

   Mostre que $T(n) \le (n+1)\lg(n+1) + 1$ para todo $n \ge 0$.

   Resolução: Prova por indução em n.

   Base: $n = 0$ e $n = 1$.

   De fato, $T(0) = 1 \le (0+1)\lg(0+1) + 1$ e $T(1) = 1 \le (1+1)\lg(1+1) + 1$.

   Passo: $n \ge 2$.

   Neste caso, temos que

$$\begin{aligned} T(n) &= \min_{0\le k\le n-1}\{T(k) + T(n-1-k)\} + n \\ &\le \min_{0\le k\le n-1}\{(k+1)\lg(k+1) + 1 + (n-k)\lg(n-k) + 1\} + n \quad \text{por indução} \\ &\le \min_{0\le k\le n-1}\{(k+1)\lg(k+1) + (n-k)\lg(n-k)\} + n + 2. \end{aligned}$$

   Considere a função $f(x) = (x+1)\ln(x+1) + (n-x)\ln(n-x)$. (Note que trocamos o lg por ln aqui, por conveniência.) Vamos determinar um ponto em que a função $f(x)$ atinge seu mínimo no intervalo $[0\,..\,n-1]$. Para tanto, vamos calcular a derivada primeira e a derivada segunda de $f(x)$. Primeiro,

$$f'(x) \;=\; \frac{x+1}{x+1} + \ln(x+1) - \frac{n-x}{n-x} - \ln(n-x) \;=\; \ln(x+1) - \ln(n-x),$$

   que vale zero quando $x + 1 = n - x$, ou seja, quando $x = (n-1)/2$. A derivada segunda é

$$f''(x) \;=\; \frac{1}{x+1} + \frac{1}{n-x},$$

   que é maior que zero para todo x em $[0\,..\,n-1]$. Assim sendo, $x = (n-1)/2$ é o único ponto de mínimo da função $f(x)$ no intervalo $[0\,..\,n-1]$.

   Observe que a função $f(k)$ difere da função que está dentro do min apenas por uma constante (o $\ln 2$), portanto o mínimo é atingido no mesmo lugar. Teríamos então que

$$\begin{aligned} T(n) &\le \min_{0\le k\le n-1}\{(k+1)\lg(k+1) + (n-k)\lg(n-k)\} + n + 2 \\ &= 2, \frac{n+1}{2}\lg\frac{n+1}{2} + n + 2 \\ &= (n+1)(\lg(n+1) - 1) + n + 2 \\ &= (n+1)\lg(n+1) - (n+1) + n + 2 \\ &= (n+1)\lg(n+1) + 1 \quad \text{como queríamos demonstrar.} \end{aligned}$$

   No entanto, há um pequeno problema neste argumento. Ao substituirmos o mínimo pelo seu valor no ponto de mínimo da função $f(x)$, ignoramos o fato de que tal mínimo foi pego apenas entre os inteiros. De fato, a primeira igualdade acima vale apenas quando $(n-1)/2$ é um inteiro, ou seja, quando n é ímpar. Quando n é par, temos que fazer algum ajuste, pois o mínimo poderia ser substituído apenas por

Por outro lado, o mínimo na expressão de $T(n)$ é tomado apenas nos inteiros do intervalo $[0\,..\,n-1]$, assim, se n é ímpar, de fato o mínimo ocorre quando $k = (n-1)/2$. Mas se n é par, o mínimo ocorre quando $k = n/2$ ou $k = (n-2)/2$. Assim sendo, $\frac{n+2}{2} \lg \frac{n+2}{2} + \frac{n}{2}$ e, infelizmente, usando esta expressão, não conseguimos chegar a mesma conclusão.

Fico por enquanto devendo um ajuste que resolva este problema. Se alguém conseguir ajustar a prova, pode mandar a sugestão para o fórum.