PUC GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
CMP1065 Projeto e Análise de Algoritmos I

Marco A. F. Menezes

AULA 10

Referências: esta aula está baseada nos seguintes livros,

1. Ruy Campello e Nelson Maculan. Algoritmos e Heurísticas: desenvolvimento e avaliação de performance. Niterói, RJ: EDUFF, 1994.

2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest e Clifford Stein. Algoritmos: teoria e prática. Tradução da segunda edição americana: Vandenberg D. de Souza. Rio de Janeiro: Campus, 2002.

3. Sanjoy Dasgupta, Christos Papadimitriou e Umesh Vazirani. Algoritmos. Tradução: Guilherme A. Pinto. São Paulo: McGraw-Hill, 2009.

4. Kenneth H. Rosen. Matemática Discreta e Suas Aplicações. Tradução: Helena Castro e João Guilherme Giudice. Sexta edição, São Paulo: McGraw-Hill, 2009.

5. Laira V. Toscani e Paulo A. S. Veloso. Complexidade de Algoritmos. Série Livros Didáticos, Número 13, Instituto de Informática da UFRGS. Segunda edição. Porto Alegre: editora Sagra Luzzatto, 2005.

6. Nivio Ziviani. Projeto de Algoritmos: com implementações em PASCAL e C. Terceira edição revista e ampliada. São Paulo: Cengage Learning, 2015.

Aulas anteriores

Unidade 2 - Paradigmas de Projeto de Algoritmos

Ideia

2.1 Divisão e conquista

A estratégia de divisão e conquista: opera decompondo um problema em subproblemas independentes, resolvendo-os e combinando as soluções obtidas em uma solução para o problema original. Isso estabelece um processo recursivo de decomposições e recombinações.

Teorema 2.1 (Teorema Mestre) Considere f uma função crescente que satisfaz a relação de recorrência

$$f(n) = af(n/b) + cn^d$$

sempre que $n = b^k$, em que k é um número inteiro positivo, $a \geq 1$, b é um número maior do que 1 e c e d são números reais, sendo c positivo e d não negativo. Então,

$$f(n) \text{ é } \begin{cases} O(n^d), \text{ se } a < b^d \\ O(n^d \log n), \text{ se } a = b^d \\ O(n^{\log_b a}), \text{ se } a > b^d. \end{cases}$$

Demonstração Considere $a \geq 1$, b é um número maior do que 1 e c e d são números reais, sendo c positivo e d não negativo. Considere $n = b^k$, em que k é um número inteiro positivo, isto é, n é uma potência de b. Pela definição da relação de recorrência,

$$f(n) = af(\frac{n}{b}) + cn^d$$

$$af(\frac{n}{b}) = a[af(\frac{n}{b^2}) + c(\frac{n}{b})^d] = a^2 f(\frac{n}{b^2}) + ac(\frac{n}{b})^d$$

$$\vdots$$

$$a^{k-1} f(\frac{n}{b^{k-1}}) = a^{k-1}[af(\frac{n}{b^k}) + c(\frac{n}{b^{k-1}})^d] = a^k f(\frac{n}{b^k}) + a^{k-1} c(\frac{n}{b^{k-1}})^d.$$

Somando ambos os lados dessas igualdades e observando que $n = b^k$, obtemos

$$f(n) = a^k f(1) + \sum_{j=0}^{k-1} a^j c(\frac{n}{b^j})^d.$$

Suponha $a = b^d$. Como $(b^j)^d = (b^d)^j$, obtemos

$$f(n) = a^k f(1) + \sum_{j=0}^{k-1} cn^d = a^k f(1) + kcn^d.$$

Usando a função logaritmo e suas propriedades,

$$n = b^k \leftrightarrow \log_b n = \log_b b^k = k \log_b b = k \leftrightarrow k = \log_b n.$$

Assim, usando o fato de que $k = \log_b n$,

$$f(n) = a^{\log_b n} f(1) + c(\log_b n)n^d.$$

Temos que $a^{\log_b n} = n^{\log_b a}$, pois tomando o logaritmo na base b em ambos os lados, $\log_b n \log_b a = \log_b a \log_b n$. Daí,

$$f(n) = n^{\log_b a} f(1) + c(\log_b n)n^d.$$

De $a = b^d$, usando logaritmo na base b em ambos os lados dessa última igualdade, temos $\log_b a = \log_b b^d = d$. Segue-se que

$$f(n) = n^d f(1) + c(\log_b n)n^d.$$

Neste ponto, se $a = b^d$, então $f(n)$ é $O(n^d \log_b n)$.

Agora, suponha que $a \neq b^d$, tal que n é uma potência de b. Vamos demonstrar por indução que

$$f(n) = c_1 n^d + c_2 n^{\log_b a}, \tag{1}$$

em que $c_1 = \frac{cb^d}{b^d - a}$ e $c_2 = f(1) + \frac{cb^d}{a - b^d}$. Considere $k = \log_b n$, em que n é uma potência de b. Passo base: se $n = 1$ e $k = 0$, então

$$c_1 n^d + c_2 n^{\log_b a} = c_1 + c_2 = \frac{cb^d}{b^d - a} + f(1) + \frac{cb^d}{a - b^d} = f(1).$$

Passo de indução: suponha que o resultado seja verdadeiro para k, em que $n = b^k$. Então, para $n = b^{k+1}$, obtemos

$$f(n) = af(\frac{n}{b}) + cn^d = a\{c_1(\frac{n}{b})^d + c_2(\frac{n}{b})^{\log_b a}\} + cn^d.$$

$$f(n) = a\{\frac{cb^d}{b^d - a}(\frac{n}{b})^d + [f(1) + \frac{cb^d}{a - b^d}](\frac{n}{b})^{\log_b a}\} + cn^d.$$

$$f(n) = \frac{acb^d n^d}{(b^d - a)b^d} + a[f(1) + \frac{cb^d}{a - b^d}]\frac{n^{\log_b a}}{b^{\log_b a}} + cn^d.$$

Uma vez que $a = b^{\log_b a}$, segue-se que

$$f(n) = n^d[\frac{ac}{b^d - a} + c] + [f(1) + \frac{cb^d}{a - b^d}]n^{\log_b a}.$$

$$f(n) = n^d[\frac{ac + c(b^d - a)}{b^d - a}] + [f(1) + \frac{cb^d}{a - b^d}]n^{\log_b a}.$$

$$f(n) = \frac{cb^d}{b^d - a}n^d + [f(1) + \frac{cb^d}{a - b^d}]n^{\log_b a}.$$

Portanto demonstramos por indução que (1) vale para $a \neq b^d$, tal que n é uma potência de b.

Neste ponto, se $a < b^d$, então pelo fato da função logaritmo ser crescente, $\log_b a < \log_b b^d = d$, de modo que o primeiro termo em (1) domina. Então, $f(n)$ é $O(n^d)$.

Por outro lado, se $a > b^d$, então pelo fato da função logaritmo ser crescente, $\log_b a > \log_b b^d = d$, de modo que o segundo termo em (1) domina. Então, $f(n)$ é $O(n^{\log_b a})$. Isso completa a demonstração.

Observação Considere o teorema mestre. Podemos reescrever a recorrência assim:

$$f(n) = af(n/b) + O(n^d).$$

Exemplo Seja a equação de recorrência

$$T(n) = 4T(n/2) + n,$$

em que $a = 4 \geq 1$, $b = 2 > 1$, $c = 1 > 0$ e $d = 1 \geq 0$. Pelo teorema mestre, uma vez que $a = 4 > 2^1 = b^d$ segue-se que $T(n)$ é $O(n^2)$, já que $\log_b a = \log_2 4 = 2\log_2 2 = 2$.

Intercalação (*merge*): é a operação de unir dois arquivos odenados gerando um terceiro arquivo ordenado.

Exercícios para casa Fazer a Prova 3.

Próxima aula Paradigmas de projeto de algoritmos: programação dinâmica.