Exercícios de Projeto e Análise de Algoritmos
Ciência da Computação
campus Foz do Iguaçu

Data: Maio/2016
Prof. Rômulo Silva

Tópicos: **Projetos de Algoritmo por Indução, Divisão e Conquista, Programação Dinâmica e Algoritmos Gulosos**

1. Suponha que seja dado um algoritmo caixa-preta (CP) de complexidade $f_{CP}(n)$ com a seguinte propriedade: dada uma sequência A de n números inteiros e um inteiro k, CP retorna verdadeiro ou falso, indicando se existe um subconjunto de números cuja soma é exatamente k. usando o algoritmo CP, projete por indução um algoritmo que retorne os elementos do subconjunto cuja soma é k. Calcule a complexidade de seu algoritmo em função de $f_{CP}(n)$.

   $CP(A, n, k)$: retorna verdadeiro se A possui subconjunto cuja soma seja exatamente k.

2. Dados dois vetores ordenados A e B de tamanhos m e n respectivamente, projete um algoritmo para encontrar o menor elemento comum aos dois vetores. Calcule a complexidade de seu algoritmo.

3. Um projeto de algoritmo usando a técnica de divisão e conquista leva a uma recorrência do tipo $T(n) = aT(\frac{n}{b}) + f(n)$ para expressar sua complexidade.

   (a) Explique o que representam a, b e $f(n)$ em relação à técnica de divisão e conquista.

   (b) Pelo teorema mestre, se $f(n) \in \Omega(n^{\log_b a+\epsilon})$ e $af(\frac{n}{b}) \leq cf(n)$ para constantes $\epsilon > 0$, $c > 1$, $a \leq 1$ e $b \leq 1$ conclui-se que $T(n) \in \Theta(f(n))$. Explique o que isto significa em termos das fases do algoritmo projetado utilizando a técnica de divisão e conquista.

4. Considere a seguinte definição: dada uma cadeia $S = \{a_1, ..., a_n\}$, $S' = \{b_1, ..., b_p\}$ é uma subcadeia de S se existem p índices $i(j)$ tal que:

   (a) $i(j) \in \{1, ..., n\}$ para $\forall j \in \{1, ..., p\}$

   (b) $i(j) < i(j + 1)$ para $\forall j \in \{1, ..., p - 1\}$

   (c) $b_j = a_{i(j)}$ para $\forall j \in \{1, ..., p\}$

   Exemplo: $S = \{ABCDEFG\}$ e $S' = \{ADFG\}$

   Projete algoritmo que dadas duas cadeias X e Y de um alfabeto Σ, determine a maior subcadeia comum de X e Y. (Dica: usar a técnica de programação dinâmica!)

5. Seja $S = \{a_1, ..., a_n\}$ conjunto de n atividades que podem ser executadas em um auditório, sendo $\forall i$, a atividade a_i começa no instante s_i e termina no instante f_i, com $0 \leq s_i < f_i < \infty$. Ou seja, supõe-se que a_i será executada no intervalo de tempo semi-aberto $[s_i, f_i)$.
   Duas atividades a_i e a_j são compatíveis se os intervalos $[s_i, f_i)$ e $[s_j, f_j)$ são disjuntos.
   Projete algoritmo para encontrar um subconjunto de atividades de S mutuamente compatíveis que tenha tamanho máximo. Suponha que $a_1, ..., a_n$ estejam ordenadas em ordem crescente de tempos de término. (Dica: usar a estratégia gulosa!)

6. Projete um algoritmo que dada uma sequência de n inteiros, retorne o valor da soma da subsequência de soma máxima. Isto é, dados $a_1, a_2, ..., a_n$, encontrar $\max\limits_{1\leq i\leq j\leq n} \left\{ \sum\limits_{k=i}^{j} a_k \right\}$. Exemplo: para o vetor $[-2, 11, -4, 13, -5, -2]$, a soma máxima é 20. Calcule a complexidade de seu algoritmo.

7. Projete algoritmo usando a técnica de programação dinâmica para calcular o número mínimo de operações de multiplicação (escalar) necessário para computar a matriz M dada por:

   $$M = M_1 \times M_2 \times ... \times M_i ... \times M_n$$

   onde M_i é uma matriz de b_{i-1} linhas e b_i colunas para $\forall i \in \{1, ..., n\}$.

   Para calcular a matriz $M' = M_i \times M_{i+1}$ são necessárias $b_{i-1} * b_i * b_{i+1}$ multiplicações entre os elementos de M_i e M_{i+1}.

Exemplo: considere vetor $b = [200, 2, 30, 20, 5]$ contendo as dimensões das matrizes M_1, M_2, M_3 e M_4. Algumas possbilidades para o produto $M = M_1 \times M_2 \times M_3 \times M_4$ são:

- $(M_1 \times (M_2 \times (M_3 \times M_4)))$: 5.300 multiplicações
- $((M_1 \times (M_2 \times M_3)) \times M_4)$: 29.200 multiplicações
- $((M_1 \times M_2) \times (M_3 \times M_4))$: 4.500 multiplicações