MAC 5711 – Análise de Algoritmos

Segundo Semestre de 2002

Segunda Prova – 21 de outubro

1. [1,5 pontos]
   Suponha que se quer determinar os k menores elementos de um vetor com n elementos distintos e não se está interessado na ordem relativa entre estes elementos. Descreva um algoritmo linear para resolver esse problema. (Não escreva pseudo-código; apenas descreva precisamente o seu algoritmo.)

2. [2,0 pontos]
   Suponha que você descobriu um algoritmo para multiplicar matrizes 4×4 que executa k multiplicações escalares (multiplicações normais, entre números reais), além de um número fixo de somas. .

   (a) Descreva um algoritmo geral de multiplicação de matrizes que se baseie em multiplicações de matrizes 4×4. Qual seria a complexidade desse algoritmo? (Nesta análise, você pode, se achar conveniente, considerar apenas matrizes quadradas cuja dimensão é uma potência de 4.).

   (b) Qual é o valor máximo de k que acarretará numa melhora sobre (a complexidade do) o algoritmo de Strassen? Justifique a sua resposta.

3. [2,0 pontos]
   Uma seqüência de n operações é executada em uma estrutura de dados. A i-ésima operação custa i se i é uma potência de 3 e 2 caso contrário. Determine o custo amortizado por operação. Apresente uma análise direta e uma análise com créditos.

4. [2,0 pontos]
   Projete uma estrutura de dados para suportar as duas operações a seguir para um conjunto S de inteiros distintos:

   $insere(S, x)$: insere x no conjunto S;

   $remove\_metade\_maior(S)$: elimina os $\lceil |S|/2 \rceil$ maiores elementos de S.

   Explique como implementar essa estrutura de dados de forma que qualquer seqüência de m operações seja executada em tempo $O(m)$. Não se esqueça, em particular, de explicar como as duas operações acima são implementadas na sua estrutura de dados e qual é o custo individual de cada uma delas.

5. [2,5 pontos]
   Uma variante do conhecido *problema da mochila* consiste no seguinte:

   > Dados dois inteiros positivos m e n e inteiros positivos v_i e w_i para cada i em $\{1, \ldots, n\}$, encontrar um subconjunto S de $\{1, \ldots, n\}$ tal que $\sum_{i \in S} w_i = m$ e $\sum_{i \in S} v_i$ é máximo.

   (a) Seja $P(i, k)$ o valor de $\sum_{j \in S} v_j$ onde S é um subconjunto de $\{1, \ldots, i\}$ tal que $\sum_{j \in S} w_j = k$ e $\sum_{j \in S} v_j$ é máximo. Se um tal S não existir, então $P(i, k) = -1$. Mostre o valor de $P(i, k)$ para $i = 0, 1, 2, 3, 4$ e $k = 0, 1, \ldots, 5$ para os seguintes dados: $m = 5$, $n = 4$, $v_1 = 1$, $v_2 = 5$, $v_3 = 3$, $v_4 = 1$, $w_1 = 3$, $w_2 = 1$, $w_3 = 2$ e $w_4 = 1$.

   (b) Escreva uma recorrência para $P(i, k)$. (Não esqueça de definir os casos base.)

   (c) Escreva um algoritmo para o problema acima. Use programação dinâmica e a recorrência do item (b). Analise o tempo comsumido pelo seu algoritmo, concluindo a sua complexidade.

   (d) (Vale um bônus de 0,5 ponto) O seu algoritmo é polinomial no tamanho da entrada do problema (ou seja, no número de bytes de um arquivo contendo os dados do problema)? Justifique a sua resposta.