MAC 5711 – Análise de Algoritmos

PRIMEIRO SEMESTRE DE 1999

Terceira Prova – 18 de junho de 1999

1. Suponha que você descobriu um algoritmo para multiplicar matrizes 4×4 que executa k multiplicações escalares (multiplicações normais, entre números reais).

   (a) **(Valor: 1.0 ponto)** Descreva um algoritmo geral de multiplicação de matrizes que se baseie em multiplicação de matrizes 4×4. Qual seria a complexidade deste algoritmo? (Nesta análise você pode, se achar conveniente, considerar apenas matrizes quadradas cuja dimensão é uma potência de 4.)

   (b) **(Valor: 0.5 ponto)** Qual é o valor máximo de k que acarretará uma melhora sobre o (a complexidade do) algoritmo de Strassen?

2. (a) **(Valor: 0.5 ponto)** Calcule a função prefixo (função Π) para o padrão *ababababbabbabaa*.

   (b) **(Valor: 1.0 ponto)** Descreva que modificações são necessárias para fazer com que o KMP encontre o comprimento do mais longo prefixo do padrão P que aparece no texto T. Justifique que, com as suas modificações, o algoritmo de fato resolve o problema acima, e diga qual é a complexidade da sua versão modificada do KMP.

3. (a) **(Valor: 1.0 ponto)** Descreva sucintamente os procedimentos de busca em profundidade e busca em largura em um grafo. Diga qual é a complexidade destes algoritmos.

   (b) **(Valor: 1.0 ponto)** Descreva um algoritmo o mais eficiente possível que, dado um grafo orientado G, verifica se G tem ou não um circuito orientado. Analise a complexidade do seu algoritmo.

4. **(Valor: 1.5 pontos)** São dados n livros $1, 2, \ldots, n$, com pesos $p_1, p_2, \ldots, p_n$, respectivamente. Os pesos satisfazem a condição $0 < p_i < 1$, para $i = 1, 2, \ldots, n$. Deseja-se acondicionar os livros em um número mínimo de envelopes satisfazendo as condições abaixo:

   (a) cada envelope contém no máximo dois livros;

   (b) em nenhum envelope o peso dos livros ultrapassa 1, isto é, livros i e j cabem no mesmo envelope se e somente se $p_i + p_j \leq 1$.

   Descreva um algoritmo cujo tempo de execução é $O(n \log n)$ que determina o número mínimo de envelopes necessários para acondicionar os livros. Justifique que o seu algoritmo de fato resolve o problema e que de fato executa em tempo $O(n \log n)$. **Dica:** use o método guloso.

5. **(Valor: 1.5 pontos)** Descreva um algoritmo o mais eficiente possível que, dado um inteiro positivo n e uma seqüência de n inteiros, determine o início e o fim de um segmento de soma máxima da seqüência.

   Exemplo: Para $n = 12$ e a seqüência $2, -3, 5, -2, 4, 1, 9, -10, 5, -2, -1, 4$, a saída do seu programa deve ser 3 e 7 (pois o segmento de soma máxima é $5, -2, 4, 1, 9$).

   Analise a complexidade de tempo e espaço do seu algoritmo. **Dica:** use programação dinâmica.

6. **(Valor: 1.5 pontos)** Considere o seguinte problema:

   SUBSET SUM:
   Instância: Um inteiro positivo B, um inteiro positivo n e uma seqüência de n inteiros positivos $p_1, p_2, \ldots, p_n$. (Você pode pensar nestes inteiros como o peso de n objetos.)
   Questão: Existe um subconjunto $A \subseteq \{1, 2, \ldots, n\}$ tal que $\sum_{i \in A} p_i = B$? (Existe um subconjunto dos objetos dados cuja soma dos pesos dá exatamente B?)

   Prove que o problema SUBSET SUM é NP-completo.

   Para isso, você pode assumir que o seguinte problema é NP-completo:
   PARTIÇÃO:
   Instância: Um inteiro positivo n e uma seqüência de n inteiros positivos $s_1, s_2, \ldots, s_n$. (Você pode pensar nestes inteiros como o peso de n objetos.)
   Questão: Existe um subconjunto $A \subseteq \{1, 2, \ldots, n\}$ tal que $\sum_{i \in A} s_i = \sum_{i \notin A} s_i$? (Existe uma partição dos objetos dados em dois subconjuntos de mesmo peso?)

   Dica: Não se esqueça de mostrar que SUBSET SUM está em NP.

BOA SORTE!!

Obs: A prova vale 9.5!