Rio de Janeiro, 4 de Maio de 2009.
PROVA 1 DE PROJETO E ANÁLISE DE ALGORITMOS
PROFESSOR: EDUARDO SANY LABER
DURAÇÃO: 3 HORAS

1. (2.0pt) Seja A um algoritmo que resolve um problema $\mathcal{P}$ e seja $T(n)$ a função que define a complexidade de pior caso de A em função do tamanho da entrada n.

a) Assuma $T(n) \in O(n^3)$. Podemos afirmar que existe $n_0 > 1$ tal que para toda entrada de tamanho n, com $n \geq n_0$, o algoritmo gasta no máximo n^3 operações? Por que?

b) Assuma $T(n) \in O(n^3)$. Podemos afirmar que existe $n_0 > 1$ tal que para alguma entrada de tamanho n, com $n \geq n_0$, o algoritmo gasta pelo menos n^2 operações? Por que?

2. (2.0pt). Proponha um algoritmo eficiente para resolver o problema abaixo e determine sua complexidade assintótica. O número de pontos é proporcional a eficiência do algoritmo proposto.

Entrada: Uma lista $S = (s_1, s_2, \ldots, s_n)$ de n números reais.

Saida: O número que aparece mais vezes em S

3. (2.0pt). Considere o seguinte procedimento para manipular uma lista A.

Proc(A)
Se $|A|$=1
    Return
Senão
    Embaralhe(A)
    Ordene(A)
    Divida A em duas listas A_e e A_d, cada uma delas com o mesmo tamanho.
    Proc(A_e)
    Proc(A_d)
Fim Proc

Assuma que o embaralhamento gaste $2n$ operações e que a divisão em A_e e A_d gaste $O(1)$ operações. Além disso, assuma que *ordene*(A) seja implementado através do HeapSort. Seja $T(n)$ o número de operações realizadas por Proc para a pior instância de tamanho n.

a) Escreva uma equação de recorrência para $T(n)$.

b) Encontre um função $f(n)$ tal que $T(n) = \theta(f(n))$.

4. (2.0pt) Seja $G = (V, E)$ um grafo não direcionado. Um triângulo em G é um conjunto de três vértices $\{u, v, w\} \subseteq V$ tal que $uv, uw, vw \in E$

Assuma que G é representado por uma lista de adjacências. Dado um vértice $v \in G$, proponha um algoritmo com complexidade $O(m)$ para testar se G tem um triângulo que contém v. Caso não consiga tal algoritmo, explicite na prova que não conseguiu e, então, proponha um outro algoritmo e analise sua complexidade.

5. (2.0pt) Responda as questões abaixo.

a) Seja G um grafo completo e não direcionado com n vértices e m arestas. Qual a altura da árvore gerada em uma busca em profundidade em G ? E por uma busca em largura?

b) Seja G um grafo direcionado com 11 vértices e 5 componentes fortemente conexas. É possível executar uma busca em profundidade em G e obter menos que 5 árvores? Por que? G pode possuir uma ordenação topológica? Por que?