Rio de Janeiro, 11 de Dezembro de 2009.
PROVA 3 DE PROJETO E ANÁLISE DE ALGORITMOS
PROFESSOR: EDUARDO SANY LABER
DURAÇÃO: 3 HORAS

1. (2.0 PT) Considere o procedimento abaixo

Alg2(A)
    Se $|A| \leq 1$ faça
        Print('Hello')
    Senão
        1. $A_1 \leftarrow |A|/2$ primeiros elementos de A
        2. Ordene os elementos de A_1 utilizando o MergeSort
        3. Alg2(A_1)

a) Escreva uma equação de recorrência para a complexidade de tempo do algoritmo Alg1.

b) Faça uma análise assintótica da complexidade de tempo do algoritmo Alg1.

2. (2.0pt) Seja $S = \{a_1, \ldots, a_n\}$ um conjunto de n números naturais distintos e um inteiro x. Considere o problema $\mathcal{P}$ de determinar se existem três números naturais distintos em S cuja soma é x

a) Seja $T(n)$ a complexidade de pior caso do algoritmo abaixo para resolver $\mathcal{P}$. Encontre $f(n)$ tal que $T(n) = \Theta(f(n))$.

Para i=1,...,n-2
    Para j=i+1,...,n-1
        Para k=j+1,...,n
            Se a_i+a_j+a_k =x
                Return SIM
Return NÃO

b) Projete um algoritmo com complexidade $O(n^2 \log n)$ para resolver o problema $\mathcal{P}$. Não é necessário apresentar o pseudo-código mas sim explicar com clareza os passos que o algoritmo deve realizar e explicar a complexidade.

3. (3.0pt). Seja $G = (V, E)$ um grafo onde cada vértice corresponde a um ponto do conjunto $\{(x, y) | x^2 + y^2 \leq 25$ e x, y são inteiros positivos$\}$.

a) Qual é altura da árvore gerada por uma DFS em G

b) Qual é altura da árvore gerada por uma BFS em G

c) Assuma agora que existe uma aresta entre dois vértices (x_1, y_1) e (x_2, y_2) **se e somente se** $|x_1 - x_2| \leq 1$ e $|y_1 - y_2| \leq 1$. Seja $d(v)$ a distância do nó v até a raiz da árvore gerada por uma BFS que começa no vértice correspondente a $(1, 1)$ Qual é o valor de $\sum_{v \in V} d(v)/|V|$?

4. (3.0PT) Seja A uma sequência de n números distintos $(a_1, \ldots, a_n)$. Seja $A' = (a_{i_1}, \ldots, a_{i_m})$ uma subsequência de A de tamanho m. A sequência A' é crescente se $a_{i_j} < a_{i_{j+1}}$ para $j = 1, ..., m - 1$. Como exemplo, se $A = (2, 6, 4, 7, 3, 1, 9)$, as subsequências $(2, 4, 7, 9)$ e $(6, 9)$ são crescentes e tem tamanhos 4 e 2, respectivamente.

a) Defina $L(i)$ como o tamanho da maior subsequência crescente de $a_1, a_2, \ldots, a_i$ que **inclui** a_i. Como exemplo, se $A = (2, 6, 4, 11, 3, 1, 15)$ então $L(7) = 4$, $L(6) = 1$ e $L(5) = 2$. Determine o valor de $L(1)$ e uma relação de recorrência que relacione $L(i)$ e $L(1), \ldots, L(i - 1)$, para $i > 1$.

b) A partir da relação obtida no item anterior, escreva o pseudo-código de um algoritmo polinomial para calcular os valores dos $L(i)$'s. Análise a complexidade deste.

c) Assuma que o vetor L já foi preenchido corretamente. Como podemos a partir de L obter o tamanho da maior subsequência crescente de A

5. (2.0pt) Seja $G = (V, E)$ um grafo não direcionado com pesos nas arestas e seja f uma aresta de G. Explique com palavras como seria um algoritmo para encontrar a árvore geradora com menor peso dentre aquelas árvores geradoras que contém a aresta f. Qual a complexidade do algoritmo?