Rio de Janeiro, 26 de Junho de 2007.

PROVA 2 DE PROJETO E ANÁLISE DE ALGORITMOS

PROFESSOR: EDUARDO SANY LABER

DURAÇÃO: 3 HORAS

1. Considere os dois procedimentos abaixo

```
Alg1(A)
    Se |A| ≤ 1 faça
        Print('Hello')
    Senão
        1. A_1 ← |A|/2 primeiros elementos de A
        2. A_2 ← |A|/2 últimos elementos de A
        3. Ordene os elementos de A_1 utilizando o MergeSort
        4. Ordene os elementos de A_2 utilizando o MergeSort
        5. Alg1(A_1)
        6. Alg1(A_2)
    Fim Se

Alg2(A)
    Se |A| ≤ 1 faça
        Print('Hello')
    Senão
        1. A_1 ← |A|/2 primeiros elementos de A
        2. Ordene os elementos de A_1 utilizando o MergeSort
        3. Alg2(A_1)
```

a) (0.5pt) Escreva uma equação de recorrência para a complexidade de tempo do algoritmo Alg1.

b) (0.5pt) Faça uma análise assintótica da complexidade de tempo do algoritmo Alg1.

c) (0.5pt) Escreva uma equação de recorrência para a complexidade de tempo do algoritmo Alg2.

d) (0.5pt) Faça uma análise assintótica da complexidade de tempo do algoritmo Alg2.

2. Uma cobertura por vértices para um grafo $G = (V, E)$ é um subconjunto de vértices $V' \subseteq V$ com a seguinte propriedade: para toda aresta $e \in E$, existe um vértice $v \in V'$ tal que v é extremidade de e. O problema da cobertura mínima consiste em encontrar a cobertura de menor cardinalidade para o grafo G. Sabe-se que este problema é NP-completo para grafos gerais. Entretanto, para árvores ele pode ser resolvido de forma eficiente. Dada uma árvore T, com raiz r, defina T_v como a subárvore de T enraizada em v. Seja $OPT(v, 1)$ a cardinalidade da cobertura com o menor número de vértices para árvore T_v, dentre as coberturas para T_v que utilizam o vértice v. Analogamente, defina $OPT(v, 0)$ como a cardinalidade da cobertura com o menor número de vértices para árvore T_v, dentre as coberturas para T_v que não utilizam o vértice v.

a) (0.6pt) Seja OPT(v) a cardinalidade da cobertura com o menor número de vértices que pode ser obtida para árvore T_v. Como podemos obter $OPT(v)$ a partir de $OPT(v, 1)$ e $OPT(v, 0)$?

b) (0.6pt) Se T_v é uma árvore com apenas um nó, quanto vale $OPT(v, 0)$ e $OPT(v, 1)$?

c) (0.6pt) Para um nó v sejam $v_1, \ldots, v_k$ os filhos de v. Encontre uma equação de recorrência que relacione $OPT(v, 1)$ com $OPT(v_1, 0), OPT(v_1, 1), \ldots, OPT(v_k, 0), OPT(v_k, 1)$. Note que nem todos os OPT´s citados precisam aparecer na equação.

d) (0.6pt) Para um nó v seja $v_1, \ldots, v_k$ os filhos de v. Encontre uma equação de recorrência que relacione $OPT(v,0)$ com $OPT(v_1,0), OPT(v_1,1), \ldots, OPT(v_k,0), OPT(v_k,1)$. Note que nem todos os OPT´s citados precisam aparecer na equação.

e) (0.6pt) Explique como obter um algoritmo polinomial para resolver o problema da cobertura mínima em árvores a partir das equações de recorrência obtidas acima.

3. Seja $U = \{o_1, ..., o_n\}$ um conjunto de itens, onde o item o_i tem peso w_i e valor v_i, para $i = 1, \ldots, n$. Assuma que os pesos e os valores são inteiros positivos. Além disso, seja uma mochila que suporta no máximo W unidades de peso, omde W é um inteiro positivo. Para o problema da mochila apresentado em sala de aula, definimos $OPT(i, P)$, para $i = 1, \ldots, n$ e $0 \leq P \leq W$, como o valor máximo que pode ser obtido colocando um subconjunto de $\{o_1, ..., o_i\}$ na mochila sem exceder o peso P.

a) (0.5pt) Assuma que $OPT(n, W) = OPT(n-1, W)$. Podemos deduzir que existe uma solução ótima que não contém o item o_n?

b) (0.5pt) Assuma que $OPT(n, W) = OPT(n-1, W)$. É possível existir uma solução ótima que contenha o item o_n ?

c) (1.0pt) Assuma que temos disponível uma tabela $M[0, .., n, 0, .., W]$, onde a entrada $M[i, P]$ foi preenchida com o valor $OPT(i, P)$, para $i = 1, \ldots, n$ e $0 \leq P \leq W$. Explique como obter, a partir desta tabela, em $O(n)$ um subconjunto de itens ótimo, ou seja, um subconjunto com peso menor ou igual a W e com valor $OPT(n, W)$.

4. (2.0pt) Seja A um vetor de n reais positivos $a_1, \ldots, a_n$. Exiba um algoritmo com complexidade $O(n \log n)$ para encontrar um par de inteiros i, j, com $1 \leq i \leq j \leq n$, que maximiza $\prod_{k=i}^{j} a_k$.

5.(2.0pt) Seja um grafo $H = (V_H, E_H)$. Um subconjunto de vértices $V' \subseteq V_H$ é dito independente se e somente se não existem arestas entre quaisquer dois vértices de V'. Assuma que na ilha de Delfos existe um oráculo que consegue resolver a seguinte questão em $O(1)$.

Dado um grafo $H = (V_H, E_H)$ e um inteiro k, existe um conjunto independente em H com tamanho maior ou igual a k ?

Dado um grafo $G = (V, E)$, explique como utilizar o oráculo acima para projetar um algoritmo com complexidade $O(|E| + |V|)$ para encontrar a cardinalidade da cobertura por vértices de cardinalidade mínima em G.