Rio de Janeiro, 01 de Dezembroo de 2010.
PROVA 3 DE PROJETO E ANÁLISE DE ALGORITMOS
PROFESSOR: EDUARDO SANY LABER
DURAÇÃO: 3 HORAS

1. Dado um grafo $G = (V, E)$, um subconjunto de vértices $V' \subseteq V$ é uma cobertura por vértices para G se e somente se para toda aresta $e \in E$, existe um vértice $v \in V'$ que é extremidade de e. Seja $w : V \mapsto Z$ uma função que associa cada vértice de G a um número inteiro positivo. O custo de uma cobertura V' é definido como $w(V') = \sum_{v \in V'} w(v)$. O problema da cobertura de custo mínimo consiste em encontrar a cobertura de menor custo em G.

Dada uma árvore T, com raiz r, defina T_v como a subárvore de T que inclui o nó v e todos os seus descendentes. Seja $OPT(v, 1)$ o custo da cobertura de custo mínimo para árvore T_v, dentre todas as coberturas para T_v que utilizam o vértice v. Analogamente, defina $OPT(v, 0)$ como a cobertura de menor custo para árvore T_v, dentre todas as coberturas para T_v que não utilizam o vértice v.

a) (0.5pt) Seja $OPT(v)$ o custo da cobertura de custo mínimo para T_v. Como podemos obter $OPT(v)$ a partir de $OPT(v, 1)$ e $OPT(v, 0)$?

b) (0.5pt) Se T_v é uma árvore com apenas um nó, quanto vale $OPT(v, 0)$ e $OPT(v, 1)$?

c) (1.0pt) Dado um nó v, sejam $v_1, \ldots, v_k$ os filhos de v. Encontre uma equação de recorrência que relaciona $OPT(v, 1)$ com $OPT(v_1, 0), OPT(v_1, 1), \ldots, OPT(v_k, 0), OPT(v_k, 1)$. Encontre também uma equação de recorrência que relaciona $OPT(v, 0)$ 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.

d) (1.0pt) Exiba o pseudo-código de um procedimento recursivo e eficiente que recebe como entrada um nó $v \in T$ e devolva $OPT(v)$. Analise sua complexidade.

2. Sejam X e Y sequências de n e m letras, respectivamente. Seja $X[p]$ a p-ésima letra de X e $Y[q]$ a q-ésima letra de Y. Uma subsequência comum de X e Y, com tamanho k, é uma sequência de k pares de inteiros $(i_1, j_1) < (i_2, j_2) < \ldots < (i_k, j_k)$ [1] tal que $X[i_a] = Y[j_a]$ para $a = 1, ..., k$. O problema da subsequência comum máxima consiste em encontrar a maior subsequência comum entre X e Y,

a) (1.5pt) Seja $OPT(i, j)$ o tamanho da maior subsequência comum entre X_i e Y_j, aonde X_i é a sequência contendo as i primeiras letras de X e Y_j é a sequência contendo as j primeiras letras de Y. Encontre uma equação de recorrência relacionando $OPT(i, j)$ com $OPT(i-1, j-1)$, $OPT(i, j-1)$ e $OPT(i-1, j)$

b) (1.5pt) Assuma que os valores de $OPT(i, j)$ já foram calculados e armazenados em uma matriz M de n linhas por m colunas. Exiba o pseudo-código de um procedimento que recebe M e devolve a maior subsequência comum entre X e Y.

3. Responda as questões abaixo

a) (0.5pt) Seja $e = (u, v)$ uma aresta arbitrária de um grafo $G = (V, E)$ e seja k um número inteiro. Mostre que existe uma cobertura de vértices para G com k vértices se e somente se pelo menos um dos grafos $G - \{u\}$ ou $G - \{v\}$ tem uma cobertura de tamanho $k - 1$.

b) (0.5pt) Mostre que um grafo com n vértices e mais que $k.(n-1)$ arestas não possui uma cobertura de tamanho k.

c) (2pt) Utilize os resultados dos dois itens acima para projetar um procedimento ExisteCobertura que recebe como entrada um grafo G e um inteiro x e devolve 'SIM' se G possui uma cobertura com x vértices e 'NÃO', caso contrário. Analise o algoritmo da melhor forma possível.

[1] $(i_a, j_a) < (i_b, j_b)$ se e somente se $i_a < i_b$ e $j_a < j_b$

4.(2.0pt) Seja $A = \{a_1, \ldots, a_n\}$ um conjunto de números reais **distintos** e seja b_i, para $i = 1, \ldots, n$, o i-ésimo menor elemento de A. Dados os inteiros K e L, com $1 \leq K < L \leq n$, descreva como seria um algoritmo eficiente para calcular $\sum_{j=K}^{L} b_j$ e analise sua complexidade. Quanto menor a complexidade computacional do algoritmo maior a pontuação.