Rio de Janeiro, 5 de Dezembro de 2011.
PROVA 2 DE PROJETO E ANÁLISE DE ALGORITMOS
PROFESSOR: EDUARDO SANY LABER
DURAÇÃO: 3:00h

1. (3.0pt) Seja T uma árvore com n nós. Dizemos que v é um nó separador de T se e somente se todas as árvores da floresta $T - v$ tem no máximo $n/2$ vértices. Sabe-se que toda árvore T tem um nó separador.

   a) Descreva com palavras como seria um algoritmo para encontrar um nó separador de T. Explique as estruturas de dados que seriam utilizadas para implementar o algoritmo e analise a complexidade do algoritmo em função de n

   Considere o seguinte procedimento:

   Proc(T:árvore)

   Se T tem mais de um nó.

   Encontre alguum nó separador v para árvore T (*)

   Para cada árvore S_i da floresta $T - v = \{S_1, \ldots, S_k\}$ faça.

   Proc(S_i)

   Fim Para

   Fim Se

   b) Seja $T(n)$ a complexidade de pior caso do algoritmo para uma árvore de tamanho n. Encontre uma equação de recorrência para $T(n)$. Assuma que a complexidade da linha (*) é aquela calculada no item (a).

   c) Assuma que todos vértices de T tem grau menor ou igual a 3. Faça uma analise assintótica de $T(n)$. Caso você não tenha conseguido fazer o item (a), assuma que a complexidade da linha (*) é $O(1)$ para resolver este item. Quanta mais justa a análise maior a pontação.

2. (3.0pt) Seja $G = (V, E)$ um grafo não direcionado. Um conjunto S de vértices é uma cobertura para o grafo G se toda aresta de E tem pelo menos uma extremidade que pertence a S. Considere o seguinte algoritmo para obter uma cobertura para G.

   $S \leftarrow \emptyset$

   Enquanto G contém arestas faça

   $v \leftarrow$ vértice de maior grau em G. Em caso de empate escolha o vértice com menor número

   $S \leftarrow S \cup v$

   $G \leftarrow G - v$.

   Fim Enquanto

   a) O conjunto S obtido ao término do algoritmo é sempre uma cobertura para G? Por que?

   b) Assuma que G é uma árvore. Mostre um exemplo em que o conjunto S obtido ao término do algoritmo é uma cobertura para G com número mínimo de vértices.

   c) Assuma que G é uma árvore. Mostre um exemplo em que S **não** é uma cobertura para G com número mínimo de vértices.

   d) Assuma que G é um caminho, ou seja, $V = \{1, \ldots, n\}$ e $E = \{\{1, 2\}, \{2, 3\}, \ldots, \{n - 1, n\}\}$. O conjunto S obtido ao término do algoritmo será sempre uma cobertura para G com o número mínimo de vértices? Por que?

3. (2.0pt) Uma empresa tem um orçamento de W reais, aonde W é um número inteiro. Sabemos que este recurso pode ser alocado entre n projetos $P_1, \ldots, P_n$ e que um investimento de $x > 0$ reais no projeto P_i produz retorno $a_i x + b_i$, aonde a_i e b_i são constantes. Note que um investimento de 0 reais não produz retorno. Além disso, sabemos que o investimento em um projeto tem que ser um número inteiro de reais. Seja $OPT(j, x)$ o retorno máximo que pode ser obtido quando temos disponível um orçamento de x reais para aplicar nos projetos $\{P_1, ..., P_j\}$.

   a) Ache uma equação de recorrência para $OPT(j, w)$. Não esqueça de indicar os casos base.

   b) Escreva o pseudo-código de um algortimo para calcular $OPT(n, W)$ e analise a sua complexidade em função de n e W.

4. (3.0pt) Sejam $X = x_1, .., x_n$ e $Y = y_1, ..., y_m$ duas cadeias de caracteres. Uma cadeia $Z = z_1, \ldots, z_k$ é uma subsequência comum de X e Y se Z é uma subsequência de X e de Y **simultaneamente**. Como exemplo, se $X = abacbad$ e $Y = adcabda$, então acd é uma subsequência comum de X e Y enquanto que $abad$ é somente uma subsequência de X.

   Seja $X_i = x_1, \ldots, x_i$ o prefixo de X de tamanho i e Y_j o prefixo de Y de tamanho j. Definimos $LCS(i, j)$ como o número de caracteres da maior subsequência comum entre $x_1, \ldots, x_i$ e $y_1, \ldots, y_j$. Sabemos que $LCS(i, j)$ satisfaz:

   - $LCS(i, j) = 0$, se $i = 0$ ou $j = 0$.
   - $LCS(i, j) = 1 + LCS(i - 1, j - 1)$, se $x_i = y_j$.
   - $LCS(i, j) = max\{LCS(i, j - 1), LCS(i - 1, j)\}$, se $x_i \neq y_j$.

   a) Descreva o pseudo-código de um algoritmo para calcular $LCS(n, m)$ e analise a sua complexidade de **tempo** e de **espaço** em função de m e n. Quanto mais eficiente maior a pontuação

   b) Assuma que temos uma matriz M de $n + 1$ linhas por $m + 1$ colunas, aonde a entrada $M[i, j]$ armazena $LCS(i, j)$. Mostre o pseudo-código de um procedimento para recuperar a maior subsequência comum entre X e Y a partir da matriz M. Analise a complexidade de tempo do pseudo-código proposto em função de m e n.

   c) **Prove** que a relação $LCS(i, j) = 1 + LCS(i - 1, j - 1)$ se $x_i = y_j$ é válida.

5. (Bônus 1.0). Explique o que você entendeu a respeito da teoria de complexidade computacional apresentada na última aula do curso. A qualidade da redação terá um peso importante na pontuação.