Rio de Janeiro, 11 de Maio de 2011
PROVA 2 DE ANÁLISE DE ALGORITMOS
PROFESSOR: EDUARDO SANY LABER
DURAÇÃO: 1:50h

1 ANULADA. (2.0pt) Seja um grafo direcionado $G = (V, E)$ e um conjunto S de k vértices de G. Explique como seria um algoritmo para decidir se existe ou não um ciclo em G que contém pelo menos dois vértices de S. Analise a complexidade do algoritmo proposto em função de m, n e k. Solucões mais eficientes terão maior pontuação.

2. (2.5pt) Um servidor recebe requisições de n clientes no instante 0, uma de cada cliente. Sabemos que o tempo que o servidor leva desde o instante em que ele começa a atender a requisição do cliente i até o instante em que ele termina é t_i. Além disso, sabemos que o servidor atende apenas uma requisição por vez e que quando uma requisição começa a ser atendida ela não pode mais ser interrompida.

a) O tempo de espera f_i de um cliente i é definido como o tempo que ele espera até que sua requisição comece a ser atendida. Proponha um algoritmo eficiente para determinar a ordem em que o servidor deve atender os clientes de modo a minimizar $(\sum_{i=1}^{n} f_i)/n$, o tempo médio de espera dos clientes. Analise a complexidade do algoritmo proposto.

b) Prove que seu algoritmo encontra a melhor solução possível.

4. (3.0) Seja $G = (V, E)$ um grafo não direcionado representado por uma lista de adjacências. Um conjunto de vértices $X \subseteq V$ é dito dominante se todo vértice v do grafo satisfaz pelo menos uma das seguintes condições: (i) $v \in X$; (ii) v é vizinho de algum vértice de X. Considere os seguinte algoritmo guloso

$X \leftarrow \emptyset$;
Enquanto G tem vértices
    $v \leftarrow$ vértice de G com maior grau
    $X \leftarrow X \cup v$
    Remova v e todos os seus vizinhos do grafo G.
Fim Enquanto
Devolva X

a) Exiba um grafo G tal que o conjunto dominante X obtido pelo algoritmo não tem cardinalidade mínima, ou seja, existe outro conjunto dominante X' para G que contem menos vértices que X.

b) Assuma que m é $O(n \log n)$. Proponha uma implementação eficiente para este algoritmo

c) Assuma que m é $\Omega(n^2)$. Proponha uma implementação eficiente para este algoritmo

d) Assuma que todo vértice de G tem grau menor ou igual a 3. O que podemos afirmar sobre o tamanho do conjunto dominante?

4. (2.0) Seja G um grafo direcionado com pesos nas arestas, seja s um vértices de G e seja k um inteiro positivo. Como podemos encontrar os k vértices mais próximos de s? Com qual complexidade? Análise em função de k, m, n.

5.(2.5pt) Responda as perguntas abaixo.

a) Em um grafo bipartido completo, todos os vértices de uma partição são ligados a todos vértices da outra. Qual é a altura da árvore gerada por uma BFS em um grafo bipartido completo? Qual é a altura da árvore gerada por uma DFS em um grafo bipartido completo?

b) Seja G um grafo direcionado e sejam u, v dois vértices de G. Assuma que ao executar uma DFS obtemos $pre(u) < pre(v) < post(v) < post(u)$. Podemos afirmar que existe um caminho entre u e v no grafo? Por que?

c) Seja G um grafo direcionado e sejam u, v dois vértices de G. Assuma que ao executar uma DFS em G obtemos $pre(u) < pre(v) < post(u) < post(v)$. O que podemos afirmar sobre a DFS ? Por que?

d) Seja G um grafo fortemente conexo. Podemos afirmar que G não adimite ordenação topológica? Por que?

e) Seja G um grafo que admite uma ordenação topológica O que podemos afirmar sobre as componentes fortemente conxas de G? Por que?