Introdução à Teoria dos Grafos
Lista de Exercícios de Algoritmos para Grafos

Bacharelado em Ciência da Computação, DCT–UFMS, 14/4/2005

Entrega em 28/04/2005

1. Dada uma representação por listas de adjacências de um pseudografo orientado G de ordem p e tamanho q, descreva um algoritmo com tempo de execução $O(p+q)$ para computar a representação por listas de adjacências do grafo "equivalente" $G' = (V_{G'}, E_{G'})$, onde $V_{G'} = V_G$ e $E_{G'}$ consiste das arestas em E_G com todas as arestas múltiplas entre dois vértices trocadas por uma única aresta e com todos os laços removidos.

2. O **quadrado** de um grafo orientado $G = (V, E)$ é o grafo $G^2 = (V, E^2)$ tal que $(u, w) \in E^2$ se e somente se para algum $v \in V$, ambos $(u, v) \in E$ e $(v, w) \in E$. Ou seja, G^2 contém uma aresta entre u e w sempre que G contém um caminho com exatamente duas arestas entre u e w. Descreva algoritmos eficientes para computar G^2 de G onde G é representado por uma lista de adjacências e por uma matriz de adjacências. Analise os tempos de execução dos seus algoritmos.

3. Argumente que na busca em largura o valor $d[u]$ atribuído a um vértice u é independente da ordem em que os vértices estão dispostos em cada lista de adjacências.

4. Forneça um algoritmo eficiente para determinar se um grafo é bipartido.

5. Mostre que a busca em profundidade de um grafo G pode ser utilizada para identificar as componentes conexas de G, e que a floresta da busca em profundidade contém a quantidade de árvores equivalente à quantidade de componentes conexas de G. Mais precisamenete, mostre como modificar a busca em profundidade de forma que a cada vértice v é atribuído um rótulo inteiro $cc[v]$ entre 1 e k, onde k é o número de componentes conexas de G, tal que $cc[u] = cc[v]$ se e somente se u e v estão na mesma componente conexa.

6. Outra forma de produzir uma ordenação topológica de um grafo acíclico e orientado G é repetidamente encontrar um vértice de grau de entrada 0, mostrá-lo na saída e remover todas suas arestas. Explique como implementar esta idéia com tempo de execução $O(p+q)$. O que acontece com este algoritmo se G contém circuitos?

7. Como pode mudar o número de componentes fortemente conexas de um grafo se uma nova aresta é adicionada?

8. Um professor acredita que o algoritmo para determinar as componentes fortemente conexas de um grafo orientado pode ser simplificado utilizando o grafo original — e não o grafo transposto — na segunda busca em profundidade, com o critério de buscar os vértices em ordem *crescente* de tempos de término. O professor está correto?