MC448 – Complexidade de Algoritmos
Lista de Exercícios 9

Orlando Lee

A numeração abaixo nos exercícios refere-se à **segunda edição** (CLRS).

Algoritmos elementares em grafos

1. (CLRS 22.1-1) Dada uma representação por listas ligadas de um grafo orientado, quanto tempo leva para se calcular o grau de saída de cada vértice? E os graus de entrada?

2. (CLRS 22.1-5) 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 $v \in V$, tanto $(u, v) \in E$ como $(v, w) \in E$. Ou seja, o grafo G^2 contém uma aresta (u, w) se em G existe um caminho de u a v com exatamente duas arestas. Descreva algoritmos eficientes para determinar G^2 a partir de G tanto para as representações por listas de adjacências como para a por matriz de adjacência. Analise a complexidade de tempo de seus algoritmos.

3. (CLRS 22.1-6) Quando uma representação por matriz de adjacência é usada, a maioria dos algoritmos requer tempo $\Omega(V^2)$, mas há exceções. Mostre que determinar se um grafo orientado G contém um **sorvedouro universal** – um vértice com grau de entrada $|V| - 1$ e grau de saída 0 – pode ser determinado em tempo $O(V)$, dada uma matriz de adjacência para G.

4. (CLRS 22.2-4) Argumente porque em uma busca em largura, o valor $d[u]$ atribuído a cada vértice é independente da ordem em que os vértices aparecem em cada lista de adjacência. Descreva um exemplo msotrando que a árvore de busca em largura obtida pelo algoritmo BFS visto em aula pode depender da ordem em que vértices aparecem nas listas de adjacências.

5. (CLRS 22.2-5) Descreva um grafo orientado $G = (V, E)$ com um vértice origem s especificado e uma árvore orientada T tal que para cada vértice $v \in V$, o único caminho de s a v em T é um caminho mais curto de s a v em G, mas T não pode ser obtida através de uma execução de BFS, independentemente da ordem em que os vértices aparecem em cada lista de adjacência.

6. (CLRS 22.3-7) Mostre um contra-exemplo para a afirmação: se existe um caminho de u a v em um grafo orientado G e se $d[u] < d[v]$ em uma busca em profundidade de G, então v é um descendente de u na floresta de busca em profundidade obtida.

7. (CLRS 22.4-2) Descreva um algoritmo linear que tem como entrada um grafo orientado acíclico $G = (V, E)$ e dois vértices s e t, e determina o número de caminhos de s a t. Note que só é preciso contar o número de caminhos e não listá-los.

8. (CLRS 22.4-3) Descreva um algoritmo que determina se um grafo não-orientado $G = (V, E)$ contém um ciclo. Seu algoritmo deve ter complexidade $O(V)$ independente de $|E|$.

9. (CLRS 22.4-5) Outra forma de obter uma ordenação topológica de uma grafo orientado acíclico $G = (V, E)$ é repetidamente encontrar um vértice com grau de entrada 0, imprimi-lo e removê-lo juntamente com todas as arestas que saem deste do grafo. Explique como implementar esta idéia em tempo $O(V + E)$.