MC558 - Projeto e Análise de Algoritmos II

Lista de exercícios 03 - Buscas em grafos, ordenação topológica

Os seguintes exercícios são para aprofundar no estudo da matéria.

1. Exercícios (Livro de texto): 22-3.1, 22-3.2, 22-3.3, 22-3.4, 22-3.5, 22-3.6, 22-3.7, 22-3.8, 22-3.9, 22-3.10, 22-3.11, 22-3.12

2. A entrada é um grafo não direcionado $G = (V, E)$, uma árvore geradora T de G e um vértice v. Projete um algoritmo para determinar se T é uma árvore de busca em profundidade enraizada em v. Em outras palavras, determine se T pode ser a saída de DFS quando as arestas estão listadas em alguma ordem e começando pelo vértice v.

3. Implemente uma busca em profundidade (DFS) usando uma pilha (de forma a eliminar a recursão). O seu algoritmo deverá *devolver uma floresta de busca em profundidade* representada por um vetor π e deve executar em tempo $O(V + E)$. Você pode utilizar as sub-rotinas de uma pilha como caixas-pretas: CriarPilha(Q), Topo(Q), Desempilhar(Q), Empilhar(Q, v).

4. Lembre-se de que o algoritmo de busca em profundidade, DFS, pode ser implementado por meio de uma pilha, ao invés de recursão. Suponha que o comprimento máximo de um caminho direcionado em um grafo direcionado G é no máximo R, isso é, todo caminho direcionado de G tem no máximo R arcos. Considere agora *sua* implementação não recursiva de DFS. Prove ou desprove que a pilha P nunca terá mais do que $R+1$ elementos na execução da chamada DFS(G).

5. Uma ponte de um grafo G é uma aresta cuja remoção incrementa o número de componentes. Dê um algoritmo de tempo $O(V + E)$ para determinar as pontes de um grafo.

6. Seja T uma árvore produzida pelo *BFS* de um grafo G. Embora os conceitos de arcos de avanço, cruzamento e retrocesso tenham sido definidos no contexto da busca *DFS*, eles fazem sentido em relação a qualquer árvore ou floresta radicada de G.

(a) Mostre que G não tem arcos de avanço em relação a T.

(b) Mostre que, se G for não-dirigido, então G não tem arcos de retorno.

(c) G pode ter arcos de cruzamento em relação a T? E se G for não-dirigido?

7. Um ponto de articulação de um grafo G é um vértice cuja remoção provoca o aumento de componentes conexas. Projete um algoritmo para encontrar pontos de articulação em tempo $O(V + E)$.

8. Uma outra maneira de obter uma ordenação topológica em um grafo direcionado $G = (V, E)$ é repetidamente encontrar um vértice com grau de entrada 0, incluí-lo na solução e depois remover do grafo esse vértice com todas as arestas de saída. Explique como implementar essa ideia de forma que o algoritmo execute em tempo $O(V + E)$.

9. Dê um algoritmo que determine se um determinado grafo não direcionado $G = (V, E)$ contém um ciclo. Seu algoritmo deve ser executado em $O(V)$, independente de $|E|$.