Universidade Estadual de Maringá
Departamento de Informática
Algoritmos em Grafos / Prof. Daniel Kikuti

Exercícios de fixação

1. Dê quatro ordenações topológicas diferentes do grafo orientado cujos arcos são: a-c a-d b-c b-d c-e d-e.

2. Escreva uma função que verifique se uma dada enumeração dos vértices de um grafo orientado é uma ordenação topológica.

3. Escreva uma função `hasCicle(G)` que dado um grafo orientado `G` devolve verdadeiro se `G` possui um ciclo ou falso caso contrário.

4. Sejam s e t dois vértices de um grafo direcionado acíclico (DAG). Se existe um caminho de s a t então mostre que não pode existir caminho de t a s.

5. Em um grafo não direcionado com dois componentes conexos, sempre é possível fazer com que o grafo fique conexo pela adição de uma única aresta. Dê um exemplo de um grafo orientado com dois componentes fortemente conexos tal que nenhuma adição de uma única aresta faça com que o grafo se torne fortemente conexo.

6. Dê um exemplo de um grafo fortemente conexo $G = (V, E)$ tal que, para cada vértice $v \in V$, removendo-se v de G resulta em um grafo que não é fortemente conexo.

7. Suponha que X e Y são dois componentes fortemente conexos de um grafo. Suponha que X e Y não são disjuntos (ou seja, que a interseção de X e Y não é vazia); mostre que $X = Y$.

8. Suponha que o grafo G consiste em um único ciclo. Quanto vale `low[v]` para cada vértice v de G?

9. Seja G uma árvore. Quanto vale `low[v]` para cada vértice v de G?

10. Execute a função GRAPHbridges para encontrar todas as pontes do grafo definido pelo seguinte conjunto de arestas.

    ```
    0-6 0-1 0-5 1-2 2-6 3-5 4-5 4-11 4-9 4-3 6-7 7-8 7-10 9-11 10-8 11-12
    ```

11. Considere o grafo com vértices `a b c d e f g h i j` definido pelas listas de adjacência a seguir.
    (a) Desenhe a floresta DFS e indique as arestas de retorno.
    (b) Determine os valores de `ord[v]` e `low[v]` para cada vértice v.

    ```
    a: b
    b: e c a
    c: j e d b
    d: e c
    e: h g f d c b
    f: g e
    g: f e
    h: j i e
    i: h
    j: h c
    ```