06 - Componentes fortemente conexos

Marco A L Barbosa malbarbo.pro.br

1. Exercício 22.5-1 do CLRS3 ou CLRS2.
2. Exercício 22.5-2 do CLRS3 ou CLRS2.
3. Exercício 22.5-3 do CLRS3 ou CLRS2.
4. Exercício 22.5-5 do CLRS3 ou CLRS2.
5. Exercício 22.5-6 do CLRS3 ou CLRS2.
6. Exercício 22.5-7 do CLRS3 ou CLRS2.

Referências

- [CLRS2] - Thomas H. Cormen et al. Introduction to Algorithms. 2nd edition. Capítulo 22.5.
- [CLRS3] - Thomas H. Cormen et al. Introduction to Algorithms. 3rd edition. Capítulo 22.5.

Exercício 3 (22.5-3)

Vamos utilizar o exemplo da figura 22.9 como um contra exemplo para mostrar que o algoritmo mais simples do Professor Bacon não funciona. No exemplo da figura 22.9 os vértices em ordem decrescente de tempo de término obtida pelo primeiro DFS é b, e, a, c, d, h, g, f, portanto, os vértices em ordem crescente de tempo de término é f, g, h, d, c, a, e, b. Quando o segundo DFS é executado, a árvore com raiz f inclui os vértices f, g e h, mas estes vértices não formam um SCC, portanto o algoritmo do Professor Bacon não funciona.

Exercício 4 (22.5-5)

A ideia é a mesma utilizada na solução do exercício 22.1-4 (converter um multigrafo para um grafo). Veja a solução disponibilizada na lista de exercícios de "Representações computacionais".

Exercício 6 (22.5-7)

Uma solução trivial é fazer um DFS iniciando em cada vértice e verificar se para cada par de vértice u e v, o vértice u é acessível a partir de v ou se o vértice v é acessível a partir de u. Este algoritmo tem tempo de execução $O(V(V + E))$. Mas é possível criar um algoritmo mais eficiente.

Considere um problema similar de identificar se para cada par de vértice u e v ou existe um caminho $u \leadsto v$ ou um caminho $v \leadsto u$ (uma dos caminhos deve existir, mas não os dois). Claramente o grafo precisa ser acíclico. Ou seja, este problema similar é identificar se um grafo orientado acíclico (gao) é semiconectado. Afirmamos que um gao $G = (V, E)$ é semiconectado se em uma ordenação topológica $v_1, v_2, \ldots, v_{|V|}$ dos vértice de G existe a aresta (v_i, v_{i+1}) para $i = 1, 2, \ldots |V| - 1$. A demostração desta afirmação ficar como exercício para o leitor. Verificar se um gao é semiconectado tem tempo de execução $O(V + E)$ (tempo da ordenação topológica mais a verificação da existência das arestas (v_i, v_{i+1})).

Para identificar se um grafo orientado $G = (V, E)$ qualquer é semiconectado, primeiro criamos o grafo de componentes fortemente conexos G^{SCC} de G e em seguida verificamos se G^{SCC} é semiconectado usando o algoritmo descrito anteriormente (que verifica se um gao é semiconectado). Vamos considerar dois vértices quaisquer u e v e os seus respectivos SCCs U e V. Se $U = V$, então existe os caminhos $u \leadsto v$ e $v \leadsto u$ e este fato não interfere no resultado do algoritmo executado em G^{SCC}. Se $U \neq V$, então, o resultado do algoritmo executado em G^{SCC} só pode ser verdadeiro se existir a aresta (U, V) (consequentemente existe um caminho $u \leadsto v$) ou (V, U) (consequentemente existe um caminho $v \leadsto u$) para todos os pares de vértices u e v e seus respectivos componentes U e V. Portanto G^{SCC} é semiconectado somente quando G é semiconectado.

O tempo de execução é $O(V + E)$, que é a soma dos tempos para criar o grafo G^{SCC} mais o tempo para verificar que G^{SCC} é semiconectado.