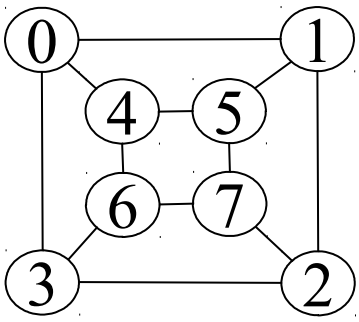
Universidade Federal do Pará
Instituto de Ciências Exatas e Naturais
Faculdade de Computação

Grafos

2ª Lista de Exercícios

Dado o grafo não direcionado **G** abaixo, responda os Itens 1 a 5.



1) Elabore um procedimento que informe a existência, ou não, de uma certa aresta no grafo de entrada. A operação deve retornar "verdadeiro", se a aresta **(i, j)** está presente no grafo, senão retornar "falso".

   Esse procedimento deve ser implementado usando matriz e lista de adjacência. Apresente *log* de execução tendo o grafo **G** como entrada.

   Agora, transforme o grafo **G** em um grafo completo **H**. É possível perceber alguma diferença entre o tempo que o procedimento leva para encontrar arestas em **H** quando se compara matriz e lista? Explique.

2) Elabore um procedimento que obtenha o conjunto de vértices adjacentes a um determinado vértice **i** presente em **G**. Esse procedimento deve ser implementado usando matriz e lista de adjacência. Apresente *log* de execução.

3) Calcule o tempo de processamento gasto pelo procedimento implementado no Item 2 para encontrar o conjunto de vértices adjacentes a cada um dos vértices de **G**. Faça isso para as duas configurações: matriz e lista de adjacência.

4) Faça um estudo detalhado acerca dos dados encontrados no Item 3. Por exemplo, que configuração foi mais rápida? Existe diferença no tempo de execução entre os vértices? Por quê?

5) Transforme o grafo **G** em um grafo completo **H**. Depois, repita os Itens 3 e 4 para o grafo **H**. O que pode ser concluído comparando os estudos realizados em **G** e **H**?

Dado o dígrafo **D**, mostrado na Figura 7.22, trabalhe os Itens 6 a 9.

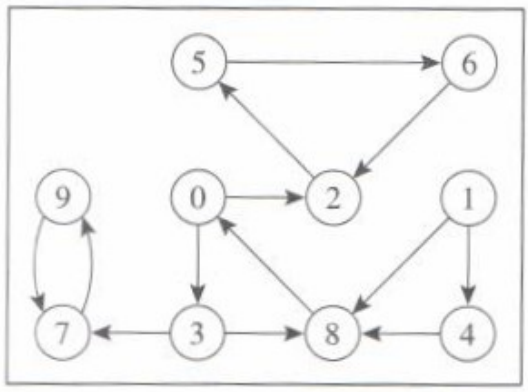


Figura 7.22 Grafo direcionado.

6) Mostre como a busca em profundidade funciona para o dígrafo **D**, ou seja, informe a ordem de visita dos vértices. Em seguida, elabore um procedimento computacional para validar o resultado encontrado. Apresente o *log* de execução.

7) Quantos e quais são os componentes fortemente conexos do dígrafo **D**? Elabore um programa computacional para validar o resultado encontrado usando busca em profundidade. Em seguida apresente um *log* com exemplo de execução.

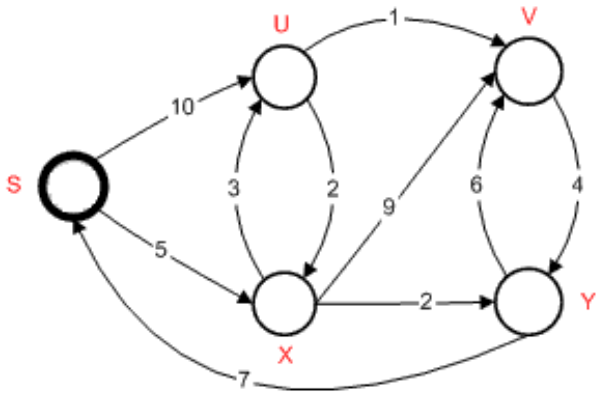
8) Diga se o dígrafo **D** possui ciclos. Em seguida, elabore um programa computacional para validar sua resposta. Por fim, apresente um *log* com exemplo de execução.

9) Dado o dígrafo **D** como entrada, elabore um programa computacional que informe o caminho mais curto entre dois dos seus vértices. Use busca em largura e apresente um *log* com exemplo de execução.

10) Elabore um programa computacional que verifique se um grafo não orientado dado como entrada é, ou não, euleriano. Se o grafo for euleriano, o programa deve ter a funcionalidade de construir um ciclo euleriano. Por fim, apresente um *log* com exemplo de execução.

11) Um vértice em um grafo não direcionado, simples e conexo é uma articulação se sua remoção torna o grafo resultante desconexo. A identificação de articulação pode ser importante em redes de computadores para identificar seus pontos frágeis. Dito isso, elabore um programa computacional usando busca em profundidade para identificar as articulações de um grafo dado como entrada. Por fim, apresente um *log* com exemplo de execução.

Dado o dígrafo **M** abaixo, trabalhe os Itens 12 a 13.



12) Implemente em uma linguagem de programação a sua escolha os algoritmos de Dijkstra e Bellman-Ford. Em seguida, use o grafo **M** como entrada do algoritmos implementados e encontre o menor caminho entre o vértice **s** e os demais vértices de **M**. Apresente o *log* de execução.

13) Modifique o valor (peso) da aresta **(x, u)** para -3. Em seguida, repita o Item 12. Explique a influência dessa modificação na execução dos algoritmos implementados.

14) Implemente em uma linguagem de programação a sua escolha o algoritmo de Prim **ou** o algoritmo de Kruskal. Em seguida, use o grafo não-orientado e ponderado abaixo como entrada do algoritmo implementado e encontre uma Árvore Geradora Mínima. Apresente o *log* de execução.