MC438/MC448 - ANÁLISE DE ALGORITMOS

IC – UNICAMP

1º Semestre de 2004

1. Construa um algoritmo linear no tamanho do grafo que dado um grafo conexo não orientado G, devolve SIM, se G possui ciclos e NÃO, caso contrário.

2. Construa um algoritmo linear no tamanho do grafo que dado um grafo conexo orientado G, testa se G é ou não acíclico.

3. Dado um grafo não orientado, proponha e analise um algoritmo para determinar se existem 2 vértices do grafo tais que não existe caminho entre eles.

4. Dê um algoritmo eficiente para determinar se um grafo é bipartido.

5. Uma k-coloração de um grafo não-orientado $G = (V, E)$ é uma função $c : V(G) \longrightarrow \{0.1.\cdots, k-1\}$ tal que $c(u) \neq c(v)$ para toda aresta $(u, v) \in E$. Os números $0, 1, \ldots, k-1$ representam as k cores, e vértices adjacentes possuem cores distintas.

   (a) Demonstre que toda árvore é 2-colorável.
   (b) Demonstre que um grafo G é bipartido se e somente se é 2-colorável.

6. Proponha um algoritmo linear no tamanho do grafo para determinar se um grafo G possui ou não um ciclo de comprimento ímpar.

7. Na *FofocaNet* existem n usuários. O usuário u fofoca com v se u manda uma mensagem para v toda vez que fica sabendo de alguma fofoca. Note que se u fofoca com v não necessariamente v fofoca com u. Você quer espalhar uma certa fofoca pela FofocaNet. Dispondo de um mapa que diz quem fofoca com quem (sendo que há m relações deste tipo), proponha e analise um algoritmo eficiente para:

   (a) Saber se existe um usuário da FofocaNet para quem você pode contar a fofoca e ter certeza de que todos os outros usuários vão ficar sabendo.
   (b) Se tal usuário existir, descobrir quanto tempo vai levar para que todo mundo da FofocaNet fique sabendo de sua fofoca, se um usuário pode mandar mensagens para vários outros ao mesmo tempo.
   (c) Saber se existe um usuário que é um túmulo: ouve fofocas criadas por qualquer usuário (ou seja: não necessariamente por aquele que contou para ele) mas não espalha a fofoca para mais ninguém.

8. Dado o grafo representado na Figura 1, determine a árvore geradora de custo mínimo usando o algoritmo de Kruskal. Desenhe cada subfloresta obtida durante a execução do algoritmo.

9. Um conjunto de n cidades está ligado por uma rede de estradas. Nessa rede as estradas são do tipo (u, v), significando que a cidade u está ligada por uma estrada à cidade v; utilizando esta estrada é possível ir de u para v ou de v para u. Por causa das chuvas de verão, algumas destas estradas ficaram *ruins* e outras ficaram *péssimas*; nenhuma permaneceu boa. É necessário interditar todas as estradas para reparos, mas isso não é possível, pois as cidades ficariam isoladas. Proponha e analise um algoritmo eficiente que descobre um jeito de manter as cidades ligadas entre si por um número *mínimo* de estradas, mas *maximizando* o número de estradas em melhor estado entre elas.

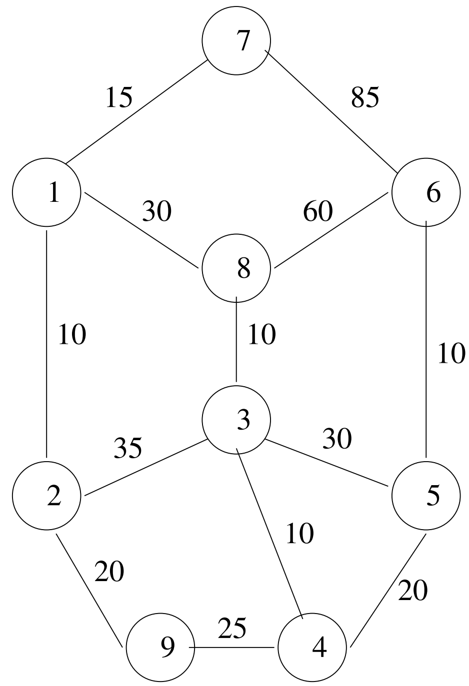

Figura 1: Grafo para o exercício 8

10. Seja $G = (V, E)$ um grafo não-orientado com custos associados as arestas e T uma árvore geradora de custo mínimo (AGM) em G. Suponha que o custo de uma única aresta e qualquer seja modificada. Diga quais as condições nas quais T deixa de ser uma AGM de G.

11. Considere o grafo dado na Figura 2.
    (a) determine as distâncias de v a cada vértice do grafo abaixo.
    (b) indique na própria figura um caminho de comprimento mínimo de v a x.
    (c) quantos caminhos desses (mínimos) existem?

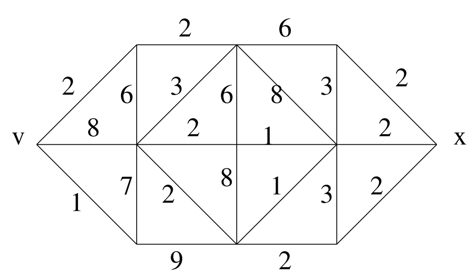

Figura 2: Grafo para o exercício 11

12. Dado um digrafo $G = (V, E)$ por sua lista de adjacências (com pesos entre parêntesis): $V = \{A, B, C, D, E, F\}$

    $A : B(4), F(2)$

    $B : A(1), C(3), D(4)$

    $C : A(6), B(3), D(7)$

    $D : A(6), E(2)$

    $E : D(5)$

    $F : D(2), E(3)$

(a) Calcule pelo método de Dijkstra o menor caminho entre C e E. Mostre a árvore determinada pelo algoritmo.

(b) Existem outros caminhos mínimos entre C e E? Quais? Qual a modificação que pode ser feita no algoritmo de Dijkstra para determinar todos os menores caminhos entre C e E?

13. Seja G um grafo orientado, cujos vértices numerados de 1 a n, já estão ordenados topologicamente. Descreva um algoritmo para encontrar um caminho de comprimento máximo em tempo linear no tamanho do grafo (i.e., $\mathcal{O}(|V| + |E|)$).

14. Escreva um algoritmo para determinar se um grafo orientado é enraizado (com a melhor complexidade de tempo que você conseguir). Um grafo é enraizado se existe um vértice $r \in V$ e existe pelo menos um caminho de r a qualquer outro vértice de G.

15. Dado um grafo não-orientado G, proponha um algoritmo de tempo polinomial para determinar se G possui um triângulo, isto é, 3 vértices mutualmente adjacentes.