Universidade Estadual de Maringá
Departamento de Informática – Bacharelado em Informática
5189 – Algoritmos em grafos / Prof. Daniel Kikuti

Aluno(a): ____

Primeira avaliação (Valor: 10,0)

1. [Valor: 2,0] Em um jogo de xadrez, podemos definir um grafo dos movimentos do bispo da seguinte forma: os vértices do grafo são as casas de um tabuleiro com N linhas e N colunas (no tabuleiro usual temos $N = 8$) e, existe uma aresta entre dois vértices se um bispo pode saltar de uma casa a outra em um só movimento (lembrando que o bispo se move sempre nas diagonais, uma ou mais casas – ver Figura 1). Considerando um tabuleiro onde $N = 3$, pede-se:
   (a) [Valor: 0,5] desenhe o grafo do bispo;
   (b) [Valor: 0,5] represente o grafo por matrizes de adjacência;
   (c) [Valor: 0,5] represente o grafo por lista de adjacência;
   (d) [Valor: 0,5] informe se o grafo é conexo e bipartido (justifique).

Figura 1: Exemplos mostrando quais casas poderiam ser alcançadas em um movimento pelo bispo B.

2. [Valor: 1,5] Para os grafos da Figura 2, apresente uma ordenação topológica válida ou justifique se não for possível definir uma ordem.

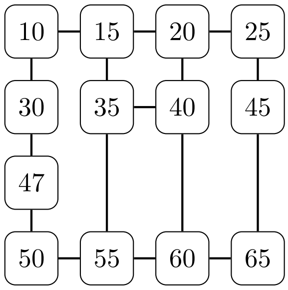
(a) (b)

Figura 2: Grafos do Exercício 2.

3. [Valor: 1,5] Sobre pontes e articulações, responda e explique seu raciocínio.
   (a) [Valor: 0,5] Dado um grafo conexo e acíclico G, quantas pontes existem em G?
   (b) [Valor: 0,5] Dada uma árvore binária completa com altura h, quantos pontos de articulação existem nesta árvore?
   (c) [Valor: 0,5] Dado um grafo que possui um ciclo hamiltoniano (isto é, um ciclo simples que passa por todos os vértices do grafo), quantas pontes e pontos de articulação existem neste grafo?

4. [Valor: 2,0] Para evitar o problema potencial de mensagens de rede (pacotes) circulando para sempre dentro de uma rede, cada mensagem inclui um campo de tempo de vida – *time to live (TTL)*. Este campo contém o número de nós (estações, computadores, etc) que podem transmitir a mensagem, encaminhando-a rumo ao seu destino, antes que a mensagem seja descartada sem cerimônia. Cada vez que uma estação recebe uma mensagem, ela decrementa o campo TTL em uma unidade. Se o destino da mensagem é a estação corrente, então o valor do campo TTL é ignorado. Entretanto, se a mensagem deve ser encaminhada, e o campo TTL decrementado contém zero, então a mensagem não é encaminhada.

   Considere a seguinte rede como exemplo:



Se uma mensagem com campo TTL igual a 2 foi enviada do nó 35 ela pode alcançar os nós 15, 10, 55, 50, 40, 20 e 60. Ela não pode alcançar os nós 30, 47, 25, 45 ou 65, visto que o campo TTL seria ajustado para zero quando a mensagem chegasse nos nós 10, 20, 50 e 60. Se aumentarmos o valor inicial do campo TTL para 3, começando do nó 35 uma mensagem poderia alcançar todos os nós, menos o 45.

Descreva um algoritmo que determine o número de nós alcançáveis, tendo como entrada uma rede, um nó inicial e um valor para o campo TTL [Problema do site UVA].

5. [Valor: 3,0] Considerando o problema de caminhos mínimos, assinale (V)erdadeiro ou (F)also.
   (a) ☐ V ☐ F O problema de caminhos mínimos possui subestrutura ótima. Ou seja, se $p = \langle s, a, b, c, t\rangle$ é um caminho mínimo de s a t que passa pelos vértices a, b e c, então é possível afirmar que não existe outro caminho de a a c cuja soma dos pesos das arestas seja menor que o caminho $p' = \langle a, b, c\rangle$.
   (b) ☐ V ☐ F A execução do algoritmo de Dijkstra no grafo da Figura 3a, tendo o vértice P como fonte, faz com que os vértices sejam removidos da fila de prioridades na ordem: P, Q, R, S, U, T.
   (c) ☐ V ☐ F Suponha que desejamos saber o caminho mínimo entre Q e S no grafo da Figura 3a. Podemos usar o algoritmo de Dijkstra começando em Q e, no momento em que S sai da fila de prioridades, podemos interromper a execução do algoritmo, pois certamente S terá a distância do caminho mínimo. Ou seja, é impossível diminuir a distância de S por outro caminho, visto que o grafo não possui arestas negativas.
   (d) ☐ V ☐ F Considerando o grafo da Figura 3b e o vértice de origem a, não podemos aplicar o algoritmo de Dijkstra visto que o grafo possui uma aresta com peso negativo. Podemos usar o algoritmo de Bellman-Ford que, após efetuar o relaxamento de cada aresta $|V| - 1$ vezes, não consegue mais relaxar nenhuma aresta e, portanto, termina com o valor correto das distâncias mínimas de a para todos os demais vértices do grafo.
   (e) ☐ V ☐ F Uma otimização para o algoritmo de Bellman-Ford é verificar se o laço interno em que cada aresta é relaxada conseguiu alterar o valor da distância estimada de algum vértice. Se após a execução do laço interno nenhum vértice teve sua distância modificada, então é porque cada vértice já possui o valor de distância mínima em relação ao vértice de origem. Portanto, nem sempre é necessário efetuar o relaxamento de cada aresta $|V| - 1$ vezes.
   (f) ☐ V ☐ F O algoritmo de Floyd-Warshall visto em aula consiste em preencher uma tabela tridimensional através da expressão $d_{ij}^{(k)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$, para k, i, j variando de 1 até n, e considerando que $D^{(0)}$ corresponde à matriz de adjacências do grafo. Este custo $O(V^3)$ de espaço de memória é desnecessário. Podemos gastar apenas $O(V^2)$ se substituirmos a expressão por $d_{ij} = \min(d_{ij}, d_{ik} + d_{kj})$.

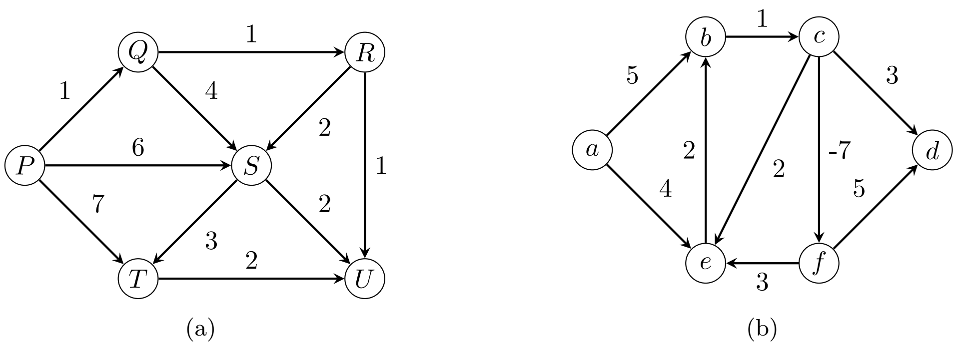


Figura 3: Grafos do Exercício 5.

Boa Prova!!!