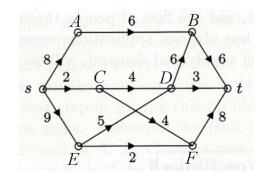
Estruturas de Dados II (DEIN0083) 2017.1 Curso de Ciência da Computação 3ª avaliação

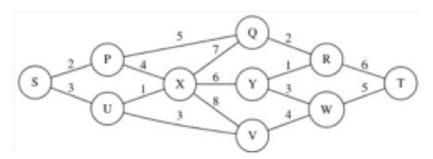
Prof. João Dallyson Sousa de Almeida			Data : 05/07/201						
Aluno:	Matrícula:								

Regras durante a prova:

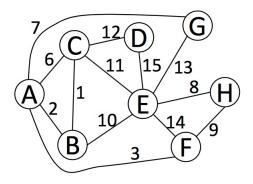
- É vetada: a consulta a material de apoio, conversa com colega e a utilização de dispositivos eletrônicos. A não observância de algum dos itens acima acarretará a anulação da prova (A).
- I. (2.0pt) Marque V para Verdadeiro e F para Falso nas seguintes afirmativas:
 - 1.() Um grafo do tipo ponderado e direcionado representado na forma de uma matriz de adjacência utiliza menor quantidade de memória que a implementação desse mesmo grafo na forma de uma lista encadeada.
 - 2.() Um grafo completo é um grafo não direcionado, no qual todos os pares de vértices são adjacentes.
 - 3.() Um grafo não-dirigido é Euleriano se não tiver nenhum vértice de grau par.
 - 4.() Os vértices dos componentes fortemente conectados de um grafo direcionado são mutuamente alcançáveis.
 - 5.() Um grafo bipartido é grafo não direcionado G = (V,E) no qual V pode ser particionado em dois conjuntos V1 e V2 tal que todas as arestas ligam os dois conjuntos V1 e V2.
 - 6.() Um dígrafo, possui um conjunto não vazio de vértices V e um conjunto de arestas E, tal que para toda aresta $(u,v) \in E$ existem mais de uma direção de u para v.
 - 7.() Um grafo G(V,E) é Hamiltoniano se existe um ciclo em G que passa por todos as arestas.
 - 8. () Uma árvore de espalhamento de um grafo ponderado conectado é mínima se a soma dos pesos de todas as arestas for mínima.
 - 9. () Um grafo é fortemente conexo se possuir mais de um componente conectado.
 - 10. () Em um grafo conectado não-dirigido e não ponderada, o caminho mais curto partindo de um nó S para todos os outros nós é calculado de forma mais eficiente, em termos de complexidade de tempo usando o algoritmo. de Floyd-Warshall.
- II. (1.0pt) Explique brevemente e mostre o resultado da execução do algoritmo de busca em profundidade na Figura da questão VI. Considere os vértices ordenados em ordem alfabética e assuma que cada lista de adjacência está ordenada alfabeticamente. Mostre o tempo e descoberta e de término de cada vértice, e mostre a classificação de cada aresta (Árvore, retorno, avanço ou cruzamento). Inicie a busca pelo vértice A.
- III. (2.0pt) Considere o grafo G abaixo. Mostre a o fluxo máximo para o grafo G, apresentando o caminho de aumento e o grafo residual em cada iteração. Considere S a fonte e T o sorvedouro. Explique o algoritmo utilizado.



IV. (1.0pt) Explique brevemente e execute o algoritmo de Dijkstra no gráfico ponderado abaixo, usando o vértice Q como a fonte. Escreva os vértices na ordem em que são marcados.



V. (2.0pt) Aplique o algoritmo de Prim começando no nó A para calcular a Árvore Geradora Mínima (AGM) do grafo abaixo. Apresente as arestas da AGM na ordem em que o algoritmo de Prim as adiciona na AGM. Use o formato (no1, no2) para apresentar as arestas adicionadas. Apresente, também, a ordem em que o algoritmo de Kruskal as adiciona na AGM. Explique brevemente os dois algoritmos.



VI. (2.0pt) Explique brevemente e execute o algoritmo do Belman-Ford no Grafo direcionado ponderado abaixo. Apresente o resultado dos vetores d e π após cada iteração.

