Prova com consulta. Duração: 2h00m. Exame Época Normal

Nome do estudante: .... **Nº**...................

Informação aos estudantes: A consulta permitida inclui slides das aulas teóricas, livros e outros materiais impressos. Não serão permitidas folhas manuscritas avulsas de qualquer tipo ou acesso à Internet (Tablets, portáteis, etc). Telemóveis deverão permanecer DESLIGADOS durante a duração do exame. Responder os seguintes grupos de questões em folhas separadas (uma folha para cada grupo): {1}, {2, 3, 4}, {5, 6}.

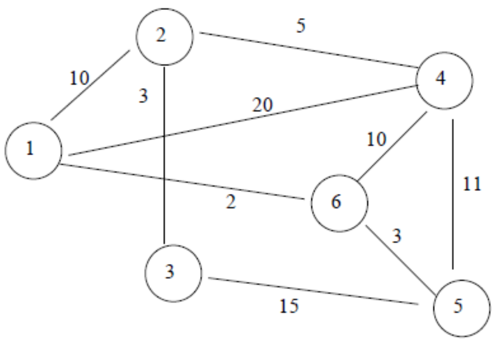
1. [**4 valores**] Considere ***D*** uma sequência de números naturais, por exemplo {3, 2, 6, 4, 5, 1}. Em tais sequências é possível identificarmos subsequências crescentes dos seus elementos, não necessariamente contíguos, tais como {3, 6}, {2, 6}, {2, 4, 5}, ou mesmo {1}. O problema da subsequência crescente máxima, (ou *Longest Increasing Subsequence*, *LIS*) pressupõe encontrar a maior (em número de elementos) das subsequências crescentes de uma dada sequência, que, no exemplo dado, será {2, 4, 5}, com três elementos. Para as alíneas seguintes, poderá utilizar pseudo-código, C++, ou Java, como achar apropriado.

a) [1,5 valores] Apresente uma solução de *Força Bruta*, e analise a sua complexidade temporal;

b) [2,5 valores] Formalize este problema utilizando a técnica de *Programação Dinâmica*, e implemente uma solução polinomial para a sua solução, indicando a respectiva complexidade temporal.

2. [**3 valores**] Considerando problemas de caminhos em grafos, responda as seguintes alíneas. Na descrição de algoritmos, procedimentos ou rotinas, utilize pseudo-código, C++, ou Java, sempre que apropriado.

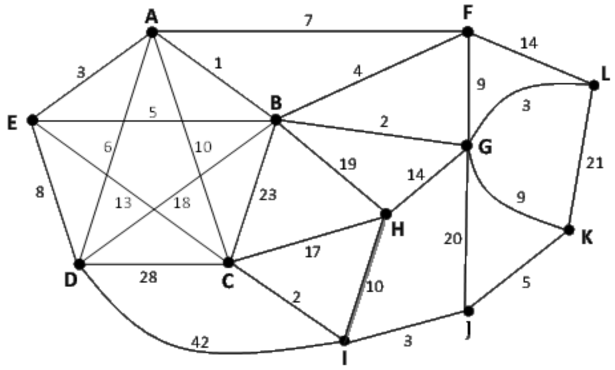
a) [1,5 valor] Considere o grafo não dirigido apresentado na figura. Calcule os caminhos mais curtos gerados pela aplicação do algoritmo de Dijkstra, sendo o nó 4 a origem dos caminhos. Apresente os valores de distância em cada vértice à medida que estes são processados.



b) [1,5 valores] Apresente um algoritmo eficiente que, dado um <u>grafo dirigido e acíclico</u> (DAG) e um vértice *v1* desse grafo, retorna o número de caminhos de *v1* para cada um dos outros vértices. Assuma que o grafo não contém arestas múltiplas (isto é, para quaisquer vértices v_x e v_y, existe, no máximo, uma aresta a ligar os dois vértices). <u>Sugestão:</u> faça uso da ordenação topológica do grafo.

3. [**4 valores**] Responda as alíneas seguintes, sobre árvores de expansão. Na descrição de algoritmos, procedimentos ou rotinas, utilize pseudo-código, C++, ou Java, sempre que apropriado.

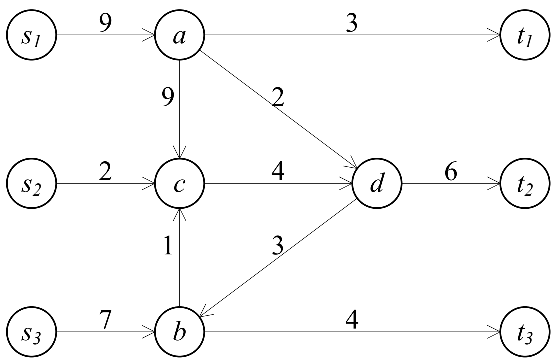
a) [2 valores] Pretende-se ligar entre si (não todas com todas) doze cidades (A, B, C, D, E, F, G, H, I, J, K, L) com novas condutas de gás a serem colocadas em estradas já existentes. O grafo seguinte apresenta as doze cidades e as estradas existentes. O peso de uma aresta representa o custo de instalar uma conduta entre as duas cidades que essa aresta liga. Determine o conjunto de arestas que minimizam o custo deste projeto. Qual o valor desse custo?



b) [1 valor] Uma árvore de expansão de *bottleneck* mínimo é uma árvore de expansão que minimiza a aresta mais pesada da árvore. Para uma árvore **T** num grafo **G**, diz-se que a aresta **e** é um bottleneck de **T** se **e** é a aresta mais pesado de **T**. A árvore **T** é uma árvore de expansão de *bottleneck* mínimo se **T** é uma árvore de expansão e não existe nenhuma outra árvore de expansão em **G** com uma aresta de *bottleneck* menos pesada. Dados um grafo **G**=(**n, m)** e um valor **w**, apresente um algoritmo que determina se existe uma árvore de expansão de *bottleneck* mínimo em que a aresta de *bottleneck* possui peso ≤ w.

c) [1 valor] Na sequência da alínea anterior (b), uma árvore de expansão de *bottleneck* mínimo do grafo **G** é sempre uma árvore de expansão mínima de **G**? Justifique a sua resposta.

4. [**3 valores**] Considere a rede de fluxos **H**, abaixo, com a indicação das suas fontes, ***S*** = {s_1, s_2, s_3}, e poços, ***T*** = {t_1, t_2, t_3}. As capacidades das arestas são dadas, próximas às respectivas arestas.



a) [1,5 valores] Encontre, realizando passo-a-passo, o fluxo máximo através da rede, e indique, para cada aresta, o seu fluxo.

b) [0,5 valor] Encontre um corte mínimo *s-t* na rede, indicando os dois subconjuntos de vértices que definem o corte mínimo.

c) [1 valor] Considere que a aresta {c, d}, por alguma razão, foi bloqueada. Que efeito terá esta situação no fluxo máximo da rede? Justifique a sua resposta.

5. [**2 valores**] Considere a seguinte mensagem, de 21 carateres, que contém 3 a's, 7 c's, 6 t's e 5 g's:

aaccccacttgggttttccgg

A seguinte sequência, de 42 bits, pode ser uma possível codificação da mesma mensagem, obtida pelo algoritmo de Huffman? Justifique a sua resposta.

000000111100010101001001001010101011100100

6. [**4 valores**] A atribuição de frequências é um problema típico em telefonia móvel: dado um número de antenas transmissoras, cada uma pode transmitir em qualquer frequência de um conjunto de frequências que lhe é atribuído. Diferentes transmissores poderão ter diferentes conjuntos de frequências. Algumas características de vizinhança entre transmissores poderão também provocar interferências, no caso de as antenas estarem a utilizar a mesma frequência (não só em termos de proximidade geográfica, mas também pela existência de prédios, colinas e outras características da rede).

Dado o conjunto de frequências de cada transmissor e os pares de transmissores que poderão causar interferência um ao outro, caso utilizem a mesma frequência, o problema da "Atribuição de Frequências" pode ser definido como: determinar se há qualquer escolha possível de frequências tal que nenhum transmissor interfira com qualquer outro, na rede de comunicação.

a) [1 valor] Reformule este problema como um problema de grafos.

b) [3 valores] Verifique se há uma solução eficiente para este problema, explicando os passos da sua solução.

Sugestão: Considere que o problema de "coloração *k*" de vértices (*vertex k-coloring*) de um grafo é NP-completo. A coloração de vértices (*vertex coloring*) pressupõe a atribuição de rótulos ou cores a cada vértice de um grafo, tal que nenhuma aresta esteja a ligar dois vértices com mesmo rótulo ou cor. A coloração de vértices de um grafo com *k* ou menos cores é conhecida por coloração-*k* (ou *k-coloring*).

Bom Exame!