

Mestrado Integrado em Engenharia Informática, MIEIC

Concepção e Análise de Algoritmos, CAL (2010-2011)

Exame com Consulta

27 de Junho de 2011

Duração: 2 horas

Nome: _____

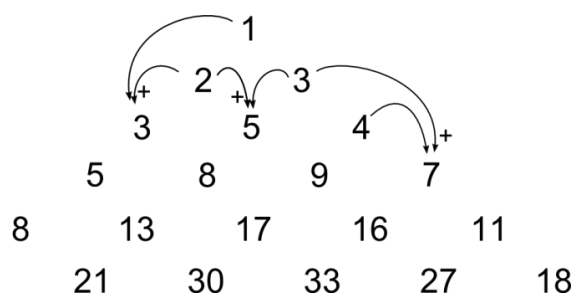
Número: _____

IMPORTANTE!

- Este exame é constituído por 6 (seis) questões, que devem ser respondidas e entregues em folhas separadas, devidamente identificadas com o nome e número do estudante; **Entregue as questões 3 e 4 numa única folha, e as demais questões em folhas distintas;**
- O enunciado deve obrigatoriamente ser entregue com as folhas de resposta, e identificado com o nome e número do estudante.

1ª Questão (4 valores): Considere o triângulo numérico da figura à direita:

À semelhança do triângulo de pascal, a construção de um elemento numa linha passa pela soma dos dois elementos superiores, com a excepção dos valores das extremidades das linhas. Para estes últimos, os valores são calculados a partir da soma dos elementos da linha anterior e anterior a essa, que se encontrem na mesma extremidade, tal como exemplificado na figura.



Implemente (em C++) a função **void triangleNumbers(int n)**, que imprime a n-ésima linha do triângulo. Utilize programação dinâmica numa abordagem com complexidade espacial linear. Por exemplo, para $n = 6$, a função imprimiria o seguinte resultado: 13 21 30 33 27 18

2ª Questão (4 valores) Considere o grafo da figura à direita, que representa a rede de estradas simplificada entre algumas cidades portuguesas:

a) [2 valores] Verifique se a rede desenhada permite definir um circuito de Euler. Em caso afirmativo, indique-o. Em caso negativo, justifique e verifique se é possível definir um caminho de Euler, identificando-o.

b) [2 valores] Um circuito hamiltoniano é definido como sendo um circuito que começa num nó de um grafo, passa por todos os restantes nós do grafo apenas uma vez e termina no nó inicial. Verifique se existe algum circuito hamiltoniano no grafo apresentado. Em caso afirmativo, indique-o. Em caso negativo, tente acrescentar uma estrada entre duas cidades para tal circuito ser possível; desenhe a estrada e indique o circuito hamiltoniano.



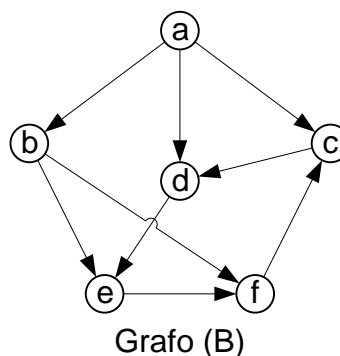
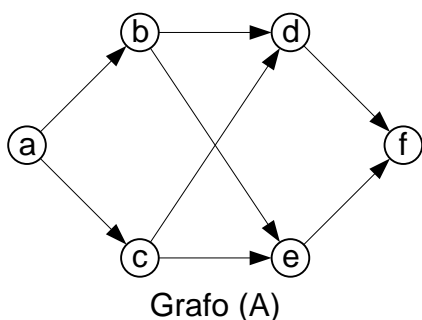
(V.S.F.F)

3ª Questão (3 valores) Considere a seguinte frase: “**pimpampumcadabolamataum**”

- [2 valores] Determine a árvore de codificação de Huffman para esta frase. Explique detalhadamente todo o processo.
- [1 valor] Usando a árvore de Huffman calculada na alínea anterior, apresente a codificação da frase “**pimpampum**”. Apresente também a codificação dos caracteres individualmente.

4ª Questão (1 valor): Entre as técnicas de pesquisa em *strings* apresentadas nas aulas, considere o algoritmo de cálculo da distância de edição para pesquisa aproximada, que pressupõe o mesmo peso para as diferentes operações (substituição, inserção e remoção de caracteres). Explique como modificaria este algoritmo para permitir dar pesos distintos às diferentes operações, indicando as fórmulas de cálculo resultantes.

5ª Questão (4 valores): Caso seja possível, defina, executando passo-a-passo o algoritmo apropriado, uma ordenação topológica para os grafos, **A** e **B**, abaixo. Caso tal ordenação não seja possível, justifique.



6ª Questão (4 valores): Considere um grafo, **G**, não dirigido, com múltiplas arestas, ciclos, e apenas pesos positivos; considere também a presença de um vértice *h*, chamado *home*. O problema da caminhada (*the Jogging problem, J*) pressupõe encontrar um percurso, de peso máximo, por exemplo, iniciando e terminando em *h*, sem repetir arestas – vértices podem ser repetidos, entretanto.

- [1 valor] Reformule o problema da caminhada, *J*, como um problema de decisão.

Considere também o problema da soma dos subconjuntos (*the Subset Sum problem, SS*), que é um problema *NP*-completo conhecido, definido da seguinte forma: dado um conjunto de inteiros positivos, *S*, há um subconjunto, $S' \subseteq S$, tal que a soma dos elementos de *S'* seja *t*?

- [1 valor] Explique, sucintamente, os passos do processo de redução de problemas da classe *NP*-completo.
- [2 valores] Prove, por redução de *SS* ($SS \geq_P J$), que *J* é *NP*-completo.

Bom Exame!