

Mestrado Integrado em Engenharia Informática, MIEIC

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

Exame com Consulta

13 de Julho 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 problema da mochila apresentado e explicado nas aulas teóricas, e que foi resolvido nas aulas práticas com programação dinâmica. Conceba, em C++, um algoritmo que imprima uma lista com os objectos a colocar **numa** mochila, de tamanho máximo n , recorrendo a recursividade e *backtracking*, garantindo uma solução óptima, mas que tenha inicialmente uma atitude gananciosa.

Dica: Tente implementar as funções partindo dos seguintes protótipos:

```
void knapsack(int n);  
KPReturn knapsack_Aux(int remainingSize, int achievedValue, vector<int>  
                      items, int addedItem);
```

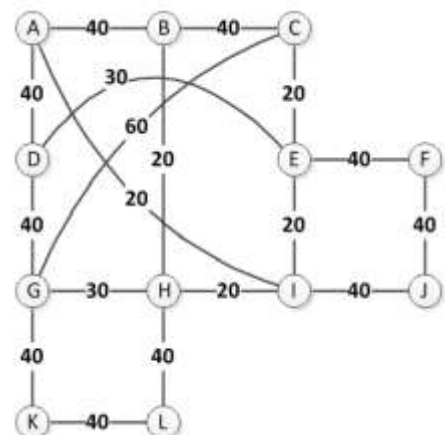
Em que KPReturn é seguinte estrutura:

```
struct KPReturn {  
    int value;  
    vector<int> items;  
};
```

2ª Questão (4 valores) Considere o grafo da figura à direita, que representa um grafo muito simplificado da rede de estradas de uma ilha – cada nó representa uma praia e cada aresta a estrada que liga duas praias, com um tempo de duração de viagem, em minutos:

a) [2 valores] Supondo que o vendedor de gelados deve fazer a sua ronda o mais rápido possível, voltando ao ponto de partida, indique um percurso ideal e o tempo gasto (pode sair de qualquer cidade).

b) [2 valores] Suponha agora que o vendedor de gelados é pago à hora e pretende ganhar o máximo possível, visitando as mesmas praias. Indique o melhor percurso possível sem repetir passagens por estradas ou praias. Qual a diferença temporal para o percurso mais rápido?



(V.S.F.F)

3ª Questão (3 valores) Pretende-se calcular a distância de edição entre as palavras “alice” e “paris”.

a) [2 valores] Construa a matriz de programação dinâmica para este problema e indique a distância de edição entre as duas palavras.

b) [1 valor] Suponha agora que não é possível substituir caracteres. Assim, a distância de edição deve ser calculada apenas com inserções e remoções. Qual a distância de edição das palavras “Alice” e “Paris”, neste caso? Explique.

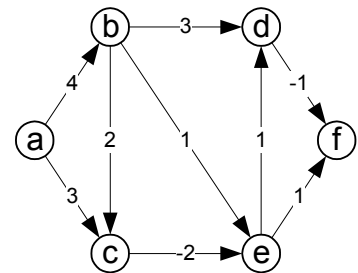
4ª Questão (2 valores): Um ficheiro contém uma mensagem de texto com m palavras diferentes, numeradas de 1 a m com ocorrências $f(1)$ a $f(m)$. Suponha que a cada palavra i está associado um custo de codificação por bit igual a $c(i)$. Atribuindo uma codificação de tamanho $l(i)$ à palavra i , o custo total de codificação é igual a $\sum(f(i) \times c(i) \times l(i))$.

Explique como modificaria o algoritmo de Huffman de modo a encontrar a codificação que conduz ao mínimo custo total.

5ª Questão (4 valores): Considere o grafo dirigido da figura ao lado, onde o número nas arestas indica os seus respectivos pesos.

a) [2 valores] Obtenha o caminho mais curto entre o vértice a e o vértice f , aplicando passo-a-passo um algoritmo adequado.

b) [2 valor] Imagine agora que se adiciona ao grafo uma aresta de f para d com peso -2. É possível calcular o caminho mais curto utilizando o algoritmo que aplicou na alínea anterior? Justifique e apresente o novo caminho mais curto, caso seja possível.



6ª Questão (3 valores): Para alguns problemas, algoritmos determinísticos não representam uma solução viável, ou por serem difíceis de formular ou por não executarem em tempo polinomial, por exemplo. Alternativas para estes casos incluem algoritmos de aproximação, algoritmos paralelos e distribuídos, ou algoritmos aleatórios. Sobre estes últimos, responda às alienas seguintes:

a) [1 valor] Em relação aos algoritmos determinísticos, “os algoritmos aleatórios sempre apresentam melhor desempenho”. Comente esta afirmação, estabelecendo uma comparação em termos de complexidade temporal entre estas duas classes de algoritmos.

b) [1 valor] As duas técnicas mais conhecidas na concepção de algoritmos aleatórios são “Monte Carlo” e “Las Vegas”. Estabeleça uma comparação entre ambas, em termos das suas eficiências temporal e espacial.

c) [1 valor] Considere um *array* A , de inteiros, e um inteiro n pertencente a este *array*. Considere ainda uma implementação para a função **finding**(*array* A , n), utilizando a técnica “Las Vegas”, que retorna a posição de n em A . Proponha, em pseudocódigo, uma implementação “Monte Carlo” equivalente, que maximiza a probabilidade da função retornar uma solução e comente a sua complexidade temporal.

Bom Exame!