AED - Algoritmos e Estruturas de Dados 2011/2012 - 1º Semestre

1º Exame, 7 Janeiro 2012, 11:30h Duração: 3 horas

Prova escrita, individual e sem consulta

NOME:____ NÚMERO:______________

PARTE I - Questões de Escolha Múltipla

Preencha as respostas na tabela (usando apenas letras maiúsculas). Se nenhuma opção servir, escreva **NENHUMA**. Se pretender alterar a sua resposta, risque e escreva ao lado a sua nova opção. Todas as questões de escolha múltipla valem 0.75 valores. As questões de escolha múltipla não respondidas são cotadas com 0 valores, mas por cada resposta errada são descontados 0.75/4 valores.

Questão	1	2	3	4	5	6	7	8
Resposta								

1. Considere o seguinte acervo ("heap")

40	37	36	32	24	33	20	19	14	17	21	30

Assumindo que a numeração da tabela começa em 0 e termina em 11, em que posições ficarão os elementos 30 (presentemente na posição 11) e 32 (presentemente na posição 3) após a chamada à função `RemoveMax(.)`?

A. Posições 3 e 1, respectivamente.	B. Posições 5 e 3, respectivamente.
C. Posições 7 e 2, respectivamente.	D. Posições 10 e 4, respectivamente.

2. Considere a seguinte recorrência

$$C_N = C_{N/3} + 2\log(N)$$

Qual dos seguintes conjuntos representa a complexidade temporal associada com a recorrência dada?

A. $\mathcal{O}(N\log(N))$ B. $\mathcal{O}(N^{3/2}\log^2(N))$ C. $\mathcal{O}(\log^2(N))$ D. $\mathcal{O}(N\log^2(N))$

3. Após a aplicação de um determinado número de iterações do algoritmo da união rápida, no problema da conectividade, obteve-se a seguinte tabela

9	8	2	0	11	2	9	12	8	9	0	11	11	5	1

De seguida aplicou-se nova iteração e obteve-se a tabela abaixo

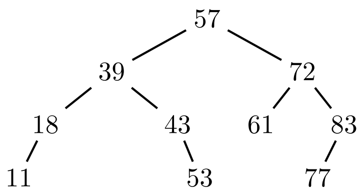
9	8	2	0	11	2	9	12	8	9	0	2	11	5	1

Considerando que os índices das tabelas são contados a partir de zero, qual das seguintes afirmações é consistente com as duas tabelas apresentadas?

A. Passaram a existir 3 conjuntos após a introdução da ligação (6, 1)
B. Existiam inicialmente 4 conjuntos e introduziu-se a ligação (7, 13)
C. Passaram a existir 4 conjuntos após a introdução da ligação (6, 10)
D. Existiam inicialmente 3 conjuntos e introduziu-se a ligação (5, 4).

4. Considere a árvore binária ordenada e balanceada AVL representada na figura.

Qual dos pares de números abaixo é tal que: a introdução do primeiro número mantém a árvore balanceada, mas a introdução do segundo requer uma rotação à direita para repor o balanceamento.



A. 29, 67.	B. 64, 41.	C. 70, 40.	D. 90, 9.

5. Indique qual das seguintes afirmações é **falsa**:

A. $\forall N > N_0, \exists c_0 : 3N^2 + 12 \leq c_0 N^2$

B. $\lim_{N\to\infty} \frac{14N^3+2N}{N^3} < \infty \;\Rightarrow\; (14N^3 + 2N) \in \mathcal{O}(N^3)$

C. $(8N^2 + 14N + \log^2 N + 2) \;\in\; \mathcal{O}(N^2)$

D. $\lim_{N\to\infty} \frac{2N^3+32N}{N^2} \to \infty \;\Rightarrow\; (2N^3 + 32N) \in \mathcal{O}(N^2)$

6. Diga qual das seguintes afirmações é **verdadeira** relativamente aos algoritmos de ordenação. Assuma em cada caso a melhor implementação de cada algoritmo (adaptativa, optimizada, etc):

A. O pior caso do Algoritmo de Inserção sucede quando os dados estão já ordenados.

B. O caso médio do Algoritmo de Selecção é idêntico ao pior caso.

C. O pior caso do Algoritmo *QuickSort* sucede quando os dados são todos idênticos entre si.

D. O pior caso do Algoritmo *BubbleSort* sucede apenas quando os dados estão já ordenados.

7. Suponha que num dado problema envolvendo uma sequência de números tem de, num dado período de tempo, efectuar um número constante de inserções (de novos números) e um conjunto linear de procuras pelo máximo, por vezes seguidas de remoção do mesmo. Podendo optar entre usar um acervo ou uma árvore binária ordenada e balanceada (BOB) em que se mantem em cada nó a altura das suas subárvores, qual das seguintes opções e justificações lhe parece mais sensata:

A. O acervo porque face à árvore BOB minimiza o tempo de inserção dos dados

B. A árvore BOB porque permite encontrar o máximo com o mesmo esforço do acervo e evita o pior caso de inserção

C. A árvore BOB pois permite minimizar a memória total ocupada

D. O acervo pois é o mais eficiente na operação mais vezes efectuada de procurar o máximo

8. Considere o grafo ponderado do Problema 10. Se aplicar o algoritmo de Kruskal para obtenção da Árvore Mínima de Suporte ("MST") e os pares de arestas indicados abaixo, qual corresponde ao caso em que a primeira entra na oitava iteração e a segunda é descartada na sexta iteração?

A. (0 – 3); (1 – 4).	B. (8 – 5); (7 – 4).	C. (5 – 8); (4 – 1).	D. (0 – 5); (2 – 7).

PARTE II - Questões de Desenvolvimento

Responda a cada uma das questões de desenvolvimento em **folhas de exame separadas** e devidamente identificadas com nome e número.

[4.0] 9. Dado um grafo ponderado não direccionado, assuma que se pretende saber se esse grafo satisfaz a desigualdade triangular. Isto é, para um qualquer par de vértices que sejam adjacentes, por exemplo o vértice v e w, pretende-se saber se para todos os vértices simultaneamente adjacentes de ambos se verificam as seguintes desigualdades:

$$\begin{aligned} d(v,w) &\leq d(v,x) + d(x,w) \\ d(v,x) &\leq d(v,w) + d(w,x) \\ d(w,x) &\leq d(w,v) + d(v,x), \end{aligned} \tag{1}$$

em que x é um vértice adjacente de v e de w e $d(i,j)$ representa o valor da aresta que liga o vérice i ao vértice j, sendo zero quando $i = j$ e ∞ quando i e j não são adjacentes. Para concretizar este objectivo propõe-se que desenvolva um conjunto de funções que a seguir se descrevem. Assuma que o grafo está representado por tabela de adjacências. Isto é,

```
typedef struct grafo {
 int V;
 int *grau;
 Aresta ** adjacentes;
} Grafo;
```

```
typedef struct {
 int vertice;
 double peso;
} Aresta;
```

O ponteiro `grau` corresponde a uma tabela indexada pelos vértices com indicação do grau de cada vértice. O ponteiro `adjacentes` corresponde a uma tabela de ponteiros para tabela de `Aresta`, indexada pelos vértices. Se um dado vértice, v, do `Grafo` G apenas possuir dois adjacentes, tem-se `G->grau[v] = 2` e a tabela `G->adjacentes[v]` tem tamanho 2. Isto é, existe `G->adjacentes[v][0]` e `G->adjacentes[v][1]`. Assuma que a estrutura com a informação relativa ao grafo já foi definida.

[1.5] a) Dado um grafo ponderado e dois vértices desse grafo, pretende-se o seguinte: se os vértices forem adjacentes, produzir a lista de todos os vértices simultaneamente adjacentes de ambos; se os vértices não forem adjacentes a lista produzida será nula, assim como será nula se não existir nenhum vértice que seja simultaneamente adjacente de ambos. A função desenvolvida por si deverá ter a seguinte assinatura:

```
Lista * GraphFindNeighbors(Grafo *G, int v, int w);
```

em que

```
typedef struct lista {
 int vertice_t;   /* vértice comum a v e w */
 double d_v_t;    /* distância de v a t */
 double d_t_w;    /* distância de t a w */
 struct lista * seguinte;
} Lista;
```

[1.5] b) Escreva agora a função que, dados dois vértices, realiza o teste da desigualdade triangular. Essa função deverá retornar 1 se todos os adjacentes comuns satisfizerem aquela desigualdade e 0 caso exista, pelo menos, um adjacente comum que a viole. A assinatura da função a desenvolver é:

```
int GraphTriangIneq(Grafo *G, Lista *L, int v, int w);
```

[0.5] c) Indique de forma precisa e justificada qual a complexidade da função desenvolvida na alínea a), como função dos parâmetros que entender adequados de entre V – número de vértices, E – número de arestas, K_i – grau do vértice i, etc..

[0.5] d) Supondo que ordena cada tabela de arestas por ordem crescente do índice dos vértices nela contidos, qual seria a complexidade da função desenvolvida na alínea a)?

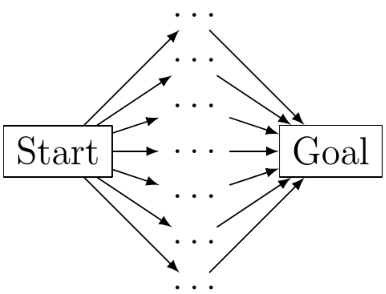
[4.0] 10. Considere um grafo ponderado não direccionado cuja matriz de adjacências se apresenta abaixo.

[3.0] a) Assumindo que a numeração dos vértices começa em zero, tome o vértice de índice 5 como ponto de partida para determinar a Árvore de Caminhos mais Curtos (“SPT”) de fonte única. Indique justificadamente todos os passos que executar e descreva em detalhe os cálculos realizados. Indique também que algoritmo usou.

$$\begin{pmatrix}
0 & \infty & 21 & 13 & 15 & 17 & \infty & \infty & \infty & \infty \\
\infty & 0 & 53 & 22 & 8 & 6 & 18 & \infty & \infty & \infty \\
21 & 53 & 0 & \infty & \infty & \infty & \infty & 18 & \infty & 4 \\
13 & 22 & \infty & 0 & 18 & \infty & \infty & 2 & \infty & 14 \\
15 & 8 & \infty & 18 & 0 & 7 & \infty & 9 & 15 & \infty \\
17 & 6 & \infty & \infty & 7 & 0 & \infty & \infty & 12 & 14 \\
\infty & 18 & \infty & \infty & \infty & \infty & 0 & \infty & 17 & 21 \\
\infty & \infty & 18 & 2 & 9 & \infty & \infty & 0 & \infty & 17 \\
\infty & \infty & \infty & \infty & 15 & 12 & 17 & \infty & 0 & 6 \\
\infty & \infty & 4 & 14 & \infty & 14 & 21 & 17 & 6 & 0
\end{pmatrix}$$

[1.0] b) Assumindo que possui um grafo representado por matriz de adjacências, discuta justificadamente qual a complexidade temporal do algoritmo de Dijkstra.

[3.0] 11. Suponha um grafo como o indicado na figura ao lado, em que foi identificado um nó inicial (*“start”*) e um outro nó terminal (*“goal”*).



O tracejado significa que o grafo não é totalmente conhecido, mas sabe-se que é “profundo”, ou seja que há muitos nós em qualquer caminho entre o nó *“start”* e o nó *“goal”*. Sabe-se igualmente que há muitos caminhos possíveis entre esses dois nós. Indique, de entre os algoritmos estudados ao longo do semestre, (DFS, BFS, etc), qual seria o mais indicado para analisar o grafo em cada um dos casos seguintes. Em cada caso e sempre que tal tal for necessário indique que estruturas de dados adicionais seria necessário utilizar (e como seriam utilizadas) para implementar o algoritmo desejado.

[1.0] a) No caso de se pretender implementar um algoritmo que encontre um caminho no grafo entre os dois nós, qualquer que seja esse caminho.

[1.0] b) Se a travessia de cada aresta tivesse um custo fixo (igual para todas) e se pretendesse encontrar o caminho mais curto entre os dois nós indicados.

[1.0] c) Se se pretender encontrar um caminho entre os dois nós que passe necessariamente por todos os nós do grafo mas apenas uma vez.

[3.0] 12. Suponha que tem uma tabela de dispersão em que as colisões são resolvidas por inserção numa lista, e onde inseriu já vários elementos. Quando inspecciona a tabela, o resultado que obtém é o seguinte:

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
↓		↓		↓		↓				↓		↓				↓		
0		21		2		33				15		26				38		
↓		↓				↓				↓		↓				↓		
30		31				63				35		6				8		
		↓														↓		
		1														58		

[1.0] a) Descreva sucintamente qual é o problema que vê.

[1.0] b) Dê um exemplo de uma função de dispersão que possa estar a resultar na distribuição indicada.

[1.0] c) Descreva uma função de dispersão melhor, i.e., que resulte numa maior dispersão de elementos.