



INSTITUTO
SUPERIOR
TÉCNICO

AED - Algoritmos e Estruturas de Dados 2009/2010 - 2º Semestre

1º Exame, 23 Junho 2010, 13:00h 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 **0.5 valores**. As questões de escolha múltipla não respondidas são cotadas com 0 valores, mas por cada resposta errada são descontados 0.5/4 valores.

Utilize a mesma tabela para indicar se respondeu ou não a cada uma das perguntas de desenvolvimento. Para tal faça uma marca para cada problema que resolveu e entregou.

Questão	1	2	3	4	5	6	7	8	9
Resposta							✓	✓	✓

1. Considere o algoritmo *quick-find* aplicado ao problema da Conectividade. Admita que o algoritmo é implementado da seguinte forma: para o par de entrada **a-b**, todos os elementos da partição de **b** são alterados para a partição de **a**. Considere que aplica este algoritmo a um problema com 10 objectos, identificados por números inteiros entre 0 e 9, com a seguinte sequência de entrada: 1-2, 3-4, 7-6, 2-6, 1-6, 4-9, 5-8

Considerando a situação **neste** momento, se na entrada aparecer **agora** o par 2 – 5, indique qual o número de elementos que mudam de partição:

A. 1	B. 2	C. 4	D. 6
------	------	------	------

2. Indique qual das afirmações seguintes é **verdadeira**:

- | | |
|----------------------------------------------|-------------------------------------------------------|
| A. $\lg^4 N \in \mathcal{O}(\lg_3 N)$ | B. $N \lg^2 N^2 \in \mathcal{O}(N^2 \lg N)$ |
| C. $\sqrt{N} \lg N \in \mathcal{O}(\lg^2 N)$ | D. $\lg \sqrt[3]{N} \in \mathcal{O}(\sqrt[3]{\lg N})$ |

3. Considere a seguinte tabela (1ª linha) sobre a qual são listados alguns passos executados por um algoritmo de ordenação (restantes linhas). Qual é o algoritmo usado?

3	10	7	11	6	1	4	9	5	2	12	8
1	3	10	7	11	6	2	4	9	5	8	12
1	2	3	10	7	11	6	4	5	9	8	12
1	2	3	4	10	7	11	6	5	8	9	12
1	2	3	4	5	10	7	11	6	8	9	12
1	2	3	4	5	6	10	7	11	8	9	12
1	2	3	4	5	6	7	10	8	11	9	12

- | |
|-------------------------------------|
| A. Inserção |
| B. <i>Shellsort</i> ($h=4, 2, 1$) |
| C. <i>Bubble</i> |
| D. <i>Quicksort</i> |

4. Considere o seguinte acervo (“heap”), no qual a prioridade é mais elevada quanto maior for o valor.

35	25	33	18	23	28	30	3	10	8	20	5	15	11
----	----	----	----	----	----	----	---	----	---	----	---	----	----

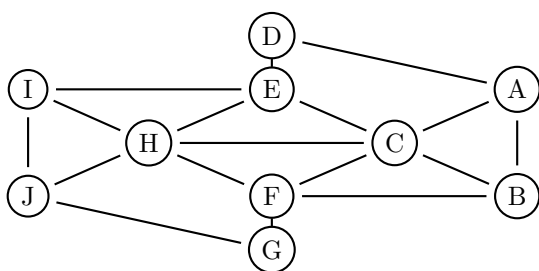
No total **quantas trocas** entre elementos da tabela são efectuadas nas rotinas de **FixUp()** e **FixDown()** durante a execução da seguinte sequência de operações (Nota: após cada operação é sempre necessário repôr a condição de acervo; conte **apenas** as trocas feitas no contexto das rotinas indicadas): (i) **Inserir** no acervo o valor 34; (ii) **Baixar** a prioridade do valor 25 para 3; (iii) **Retirar** o elemento mais prioritário.

A. 5	B. 6	C. 8	D. 9
------	------	------	------

5. Suponha que os seguintes números são introduzidas numa tabela de dispersão ("hash table"): **14735, 28342, 53735, 35, 1300, 2342, 31753, 41735, 28342, 20035, 18645, 19735**. Assuma que a função de dispersão (para indexação na tabela) é a seguinte: $f(x) = x \bmod 1000$ (ou seja, o resto da divisão por 1000). Assuma que números repetidos não são armazenados, quando tal é verificado, e que colisões são resolvidas por lista, com inserção no **final**. Qual das afirmações é **verdadeira**? (contabilize comparações apenas entre números)

- | | |
|------------------------------------------|----------------------------------------|
| A. Ocorrem 7 colisões e 9 comparações. | B. Ocorrem 6 colisões e 9 comparações. |
| C. Ocorrem 10 colisões e 10 comparações. | D. Ocorrem 6 colisões e 8 comparações. |

6. Considere o grafo indicado em baixo à esquerda e assuma que o mesmo não é direccionado mas é ponderado, como se indica do lado direito do grafo. Assumindo que aplica o algoritmo de Prim tomando o vértice **D** como ponto de partida, indique qual das afirmações é **falsa**.



A↔D, B↔F, E↔H	2
A↔B, C↔F	3
A↔C, D↔E, G↔J	4
B↔C, C↔E	5
E↔I	6
C↔H, I↔J	7
H↔I, F↔H	8
H↔J, F↔G	9

- | |
|---------------------------------------------------------------------------------|
| A. O custo da MST que se obtém é 33 e tem 9 arestas. |
| B. A aresta que une os vértices D e E é a quinta a entrar na MST. |
| C. A aresta que une os vértices C e F é a quarta a entrar na MST. |
| D. A aresta que une os vértices H e I é a oitava a entrar na MST. |

PARTE II - Questões de Desenvolvimento

Responda às questões de desenvolvimento em folhas de exame devidamente identificadas com nome e número. Responda a cada questão numa folha separada.

- [4.0] 7. A sua empresa foi contratada para resolver um problema aparentemente complicado que envolve um determinado cliente. Como teve AED no seu curso, você apercebe-se de que o problema pode ser descrito como um grafo e que a solução passa por determinar se há ou não algum ciclo nesse grafo. Reconhecendo que esta é uma excelente oportunidade para mostrar as suas capacidades e conhecimentos você oferece-se de imediato para realizar esta tarefa.

- [2.5] a) Escreva o código de uma função `detect_cycle()`, cuja assinatura é a indicada em baixo à direita e que implementa um algoritmo por si projectado para **verificar a existência de ciclos** no grafo. A função deve retornar 0 se não existirem ciclos no grafos ou um valor inteiro maior do que 0 que indica o comprimento do ciclo encontrado, medido pelo número de arestas do ciclo. Por exemplo o grafo do Problema 6 tem um ciclo de tamanho 4 constituído pelas arestas que ligam os nós **C-E-H-F** (além de vários outros ciclos).

Admita que a representação do grafo se suporta na estrutura indicada em baixo à esquerda e que foi já *alocada* e preenchida com os dados do grafo. Assuma que além desta representação, **G**, existe uma tabela de inteiros, `marked[]`, cuja dimensão é o número de nós do grafo, $G \rightarrow V$, e que foi inicializado a 0 em todas as posições.

```
typedef struct Dgraph {
    int V;
    int E;
    int **adj;
} DGRAPH;
```

```
DGRAPH *G;
int marked[];

int detect_cycle(DGRAPH *G, int marked[]);
```

Sugestão: Para facilitar a compreensão do seu código e garantir que pequenos erros não limitam a avaliação do mesmo, é conveniente descrever de forma muito sucinta o algoritmo que o código implementa.

Nota: Apenas se pretende determinar se há ou não há algum ciclo, qualquer que seja o seu comprimento, pelo que assim que se descobrir um ciclo a função deve retornar imediatamente o comprimento desse ciclo e terminar a procura.

- [0.75] b) Determine e justifique qual a complexidade do algoritmo que implementou na função da alínea a). Apresente o seu resultado em função dos parâmetros do grafo ($G \rightarrow V$, $G \rightarrow E$, etc).
- [0.75] c) Suponha agora que se pretende determinar não apenas se existe um ciclo no grafo, mas se existe um ciclo constituído por todas as arestas do grafo (sem repetir nenhuma aresta). **Sem escrever qualquer código** diga como poderia determinar se tal ciclo existe? Se existe tal ciclo, como se designa? Se existe tal ciclo, qual é o seu comprimento, em termos dos dados da representação do grafo ($G \rightarrow V$, $G \rightarrow E$, etc)?

Mude de folha para responder à pergunta 8

- [2.0] 8. Apresentando todos os cálculos, resolva a recorrência seguinte e determine a ordem da respectiva solução utilizando a notação assintótica estudada ($C_N \in \mathcal{O}(?)$):

$$C_N = C_{N/3} + \lg_3 N$$

Nota: a sua resposta deve corresponder ao menor dos majorantes.

Mude de folha para responder à pergunta 9

- [3.0] 9. Considere de novo o grafo do Problema 6.
- [2.0] a) Determine a árvore de caminhos mais curtos (SPT) tomando o vértice **A** como fonte. Apresente os seus cálculos de forma clara, detalhada e completa para cada iteração do algoritmo. Por exemplo, mas sem se restringir a estes aspectos, identifique em cada passo do algoritmo o estado da franja de procura e os respectivos pesos, assim como por que ordem entra cada vértice na árvore. Deverá ainda indicar qual o valor final do vector st (que codifica a SPT) e do vector wt (que contém as distâncias dos vértices ao vértice fonte).
- [0.5] b) Indique quais os vértices que fazem parte do caminho mais curto entre o vértice **A** e o vértice **J**, bem como qual o seu valor de custo.
- [0.5] c) Diga se a SPT que calculou lhe permite ou não dizer qual o caminho mais curto entre **C** e **F**. Porquê?