UNIVERSIDADE AbERTA www.uab.pt

Ministério da Ciência, Tecnologia e Ensino Superior

U.C. 21046

Estruturas de Dados e Algoritmos Fundamentais

08 de julho de 2019

INSTRUÇÕES

- Leia estas instruções na totalidade antes de iniciar a resolução da prova.
- O enunciado da prova é constituído por 5 grupos de questões e termina com a palavra FIM.
- Se o seu exemplar não estiver completo ou nele se verificar qualquer outra deficiência, por favor dirija-se ao professor vigilante.
- A prova deve ser resolvida na sua totalidade em folhas de respostas.
- Nas respostas, tenha a preocupação de utilizar uma letra legível.
- Todas as respostas devem ser escritas unicamente com caneta azul ou preta.
- O teste é SEM CONSULTA. Todos os elementos necessários à resolução são fornecidos no enunciado.
- É permitido utilizar máquina de calcular.
- As cotações são indicadas por grupo e nas próprias questões.
- Nas questões de escrita de programas, a sua correção terá em conta critérios de proficiência e compreensibilidade do código (legibilidade, indentação, estrutura, comentários e explicação geral).
- O não cumprimento das instruções implica a anulação das respetivas questões.
- O tempo de realização da prova é de 150 minutos.

Grupo I [3 valores]

1.1. [1] Utilizando a definição, prove que $f(n) = 2n^3 + 2n + 30$ é $\Omega(n^3)$.

1.2. Para cada um dos seguintes pares de funções $f(n)$ e $g(n)$, indique se $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$, $f(n) = \Theta(g(n))$ ou nenhum dos casos.

1.2.1. [0.5] $f(n) = \sqrt{n} + 1, \;\; g(n) = \log_2 n^2 + 1000$

1.2.2. [0.5] $f(n) = n^3 + 10n^2, \;\; g(n) = 2^{\sqrt{n}} + n^2$

1.3. [1] Considere a complexidade do seguinte segmento de código em termos do nº $f(n)$ de operações aritméticas realizadas na variável a. Determine a expressão de $f(n)$ e indique a sua complexidade na notação $O(.)$.

```
for(a=0,i=4; i<=n; i*=2)
   for(j=1; j<=i; ++j)
      a++;
```

Grupo II [5 valores]

2.1. [2] Considere uma árvore binária do tipo min Heap inicialmente vazia na qual foram inseridas as chaves 6, 9, 7, 3, 2, 1, 8, 5 pela ordem indicada. Indique na forma de vetor o Heap após cada inserção (apresente 8 vetores).

2.2. [1] Remova do Heap obtido na alínea anterior a chave de menor valor. Indique na forma de vetor o Heap resultante.

2.3. [2] Considere uma árvore B (B-Tree) de ordem 3 inicialmente contendo apenas o nó raiz com as chaves 12, 5. Desenhe a árvore após cada uma das seguintes operações de inserção (I) e remoção (R) pela ordem indicada: I 8, 7, 14, 18, 20; R 8; (total de 6 desenhos).

Grupo III [4 valores]

3. Considere o vetor [8 4 3 1 2 6 9 5]. Ordene o vetor utilizando o algoritmo de ordenação indicado. Explique de um modo geral o funcionamento do algoritmo e indique a sequência de vetores/passos correspondentes às iterações principais do algoritmo.

3.1. [2] Algoritmo de ordenação por fusão (Merge Sort).

3.2. [2] Algoritmo de ordenação Comb Sort.

Grupo IV [4 valores]

4. Pretende-se conceber em C++ uma estrutura de dados tipo lista duplamente ligada (double linked list) em que os itens são inteiros. A lista deve suportar as operações de: (i) Remover e retornar o item do fim da lista (método delete_end); (ii) Inserir um item numa posição da lista (método insert_pos).

Na resolução das alíneas seguintes pode criar métodos e construtores adicionais que achar convenientes. Explique em termos gerais o funcionamento do código e indique explicitamente situações especiais ou casos particulares considerados.

4.1. [1] Defina as classes que entender necessárias para a implementação da lista. Defina apenas atributos e métodos, não inclua código para os métodos. Justifique cada atributo que incluir nas classes.

4.2. [1] Implemente o método "delete_end".

4.3. [2] Implemente o método "insert_pos".

Grupo V [4 valores]

5. Pretende-se conceber em C++ uma estrutura de dados tipo àrvore de pesquisa binária (BST - Binary Search Tree) em que os itens são genéricos. A BST deve suportar as operações de: (i) Efetuar a travessia da árvore em ordem (método inorder); (ii) Remover um item da árvore utilizando o algoritmo de remoção por fusão (método deleteByMerging).

Na resolução das alíneas seguintes pode criar métodos e construtores adicionais que achar convenientes. Explique em termos gerais o funcionamento do código e indique explicitamente situações especiais ou casos particulares considerados.

5.1. [1] Defina as classes que entender necessárias para a implementação da BST. Defina apenas atributos e métodos, não inclua código para os métodos. Justifique cada atributo que incluir nas classes.

5.2. [1] Implemente o método "inorder". Visitar um nó significa imprimir o item do nó.

5.3. [2] Implemente o método "deleteByMerging". O método deve retornar 0 em caso de sucesso e -1 se o item indicado não for encontrado.

FIM