UNIVERSIDADE ABERTA
www.uab.pt

Ministério da Ciência, Tecnologia e Ensino Superior

U.C. 21046

Estruturas de Dados e Algoritmos Fundamentais

06 de setembro 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 $f(n) = \log_2 n + \sqrt{n} + 10$ é $O(\sqrt{n})$.

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] $f(n) = n^{3/2} + n^{\sqrt{n}}$, $g(n) = n^3 + 1000$

1.2.2. [0.5] $f(n) = 10 \log n$, $g(n) = \log n^2$

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=1; i<=n*n; ++i)
   for(j=i; j<=n*n; ++j)
      a++;
```

Grupo II [5 valores]

2.1. Considere uma árvore de promoção (Splay Tree) inicialmente vazia.

2.1.1. [1] Insira na árvore as chaves 3, 12, 8, 9, 17, 13, 20, 1, 4 pela ordem indicada. Desenhe a árvore obtida após efetuadas todas as inserções (total de 1 desenho). Nas alíneas seguintes considere a árvore obtida como a árvore "original".

2.1.2. [1] Remova da árvore original a chave 12 utilizando o algoritmo de remoção por fusão (Deletion by Merging), escolhendo a opção em que o nó removido é substituído pelo seu filho esquerdo. Desenhe a árvore obtida.

2.1.3. [1] Considere que na árvore original foi efetuado um acesso à chave 9. Desenhe a árvore obtida após o acesso (total de 1 desenho).

2.2. [2] Construa uma função recursiva que dado um vetor v com n elementos calcule a soma de todos os seus elementos pares. O protótipo da função é dado por `int soma(int v[], int n);`.

Grupo III [4 valores]

3.1. [2] Considere uma tabela T de dispersão (hash) com dimensão 9 e função de hash $h(x) = x \bmod 9$. As colisões são resolvidas com sondagem (probing) linear. Considerando a tabela inicialmente vazia, determine o conteúdo final da tabela após a inserção das chaves 1, 13, 10, 8, 9, 18, 17 pela ordem apresentada. Justifique os cálculos efetuados para cada inserção.

3.2. [2] Considere o vetor [9 4 5 8 3 6 1 7 2]. Indique a sequência de passos para a sua ordenação utilizando uma versão simplificada do algoritmo QuickSort (sem questões de eficiência na troca de elementos para a divisão dos vetores). Utilize para pivot o elemento do meio. Justifique de um modo geral o funcionamento do algoritmo.

Grupo IV [4 valores]

4. Pretende-se conceber em C++ uma estrutura de dados tipo lista simplesmente ligada (single linked list) em que os itens são inteiros. A lista deve suportar as operações de: (i) Inserir um item no início da lista (método insert_0); (ii) Remover e retornar o item do fim da lista (método delete_end).

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 "insert_0".

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

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) Inserir um item na árvore (método insert); (ii) Remover um item da árvore utilizando o algoritmo de remoção por cópia (método deleteByCopying).

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 "insert".

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

FIM