Departamento de Engenharia Informática
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA
PÓLO II – Pinhal de Marrocos
3030-290 Coimbra, Portugal

Algoritmos e Estruturas de Dados

Exame de Treino

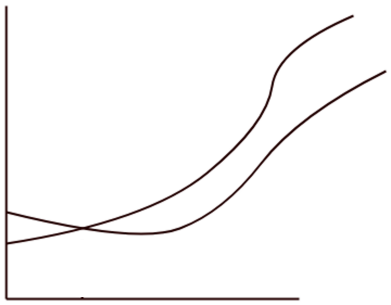
Nome: **N. Est./St ID:**

email: **Avaliação para 50 pontos**

Pontuação Exame: **Pontuação Aval. Cont. AED T:** **Pontuação Exame (0..50):**

GRUPO A – Análise de Complexidade e Técnicas de Desenho de Algoritmos

A.1 (10 pontos) Considere a seguinte figura e ilustre os elementos relevantes para representar a notação de complexidade O-grande



Descreva agora sucintamente o que representa dizer que um algoritmo tem complexidade temporal $O(N^2)$. Seja coerente com letras usadas na ilustração acima:

A.2 (10 pontos, cada afirmação incorreta penaliza em 5 pontos) Considere as técnicas de desenho de algoritmos estudadas. Quais das seguintes afirmações são verdadeiras:

☐ a programação dinâmica é uma técnica de transformação de um algoritmo recursivo em iterativo ☐ a programação dinâmica tende a criar uma recursão mais curta (menor profundidade da árvore de recursão) ☐ a recursão indireta pode ser transformada numa iteração sem recurso a pilha auxiliar ☐ a recursão terminal beneficia de um método que guarde resultados intermédios ☐ a recursão terminal pode facilmente ser convertida numa estrutura iterativa

Justifique sucintamente as seleções feitas acima como verdadeiras. Explique sucintamente os conceitos envolvidos (ex.: recursão indireta, programação dinâmica):

GRUPO B – Estruturas de Dados

B.1 (5 pontos) Considere uma Árvore VP. Indique as propriedades a que obedece uma árvore deste tipo:

Prop. #1 _____

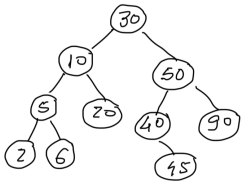
Prop. #2 _____

Prop. #3 _____

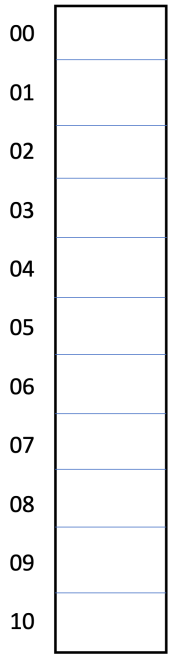
Prop. #4 _____

Prop. #5 _____

B.2 (20 pontos) Considere a árvore AVL representada abaixo. Mostre a evolução da árvore quando elimina sequencialmente os seguintes elementos: **40 50 45**

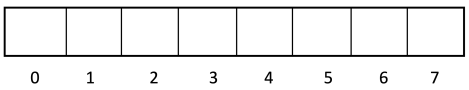


B.3 (15 pontos) Considere o espaço de memória abaixo. Crie uma tabela de dispersão que utilize todo o espaço de memória disponível. Use como função de dispersão o resto da divisão e como função de dispersão *quadratic probing*. Faça as parametrizações que achar necessárias e adequadas. Mostre o resultado da inserção das seguintes chaves **21 10 32 18 19 20 29.** Apresente os cálculos efetuados.

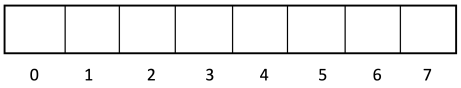


GRUPO C – Algoritmos de Ordenamento

C.1 (20 pontos) Considere o algoritmo de ordenamento *HeapSort* usado para colocar por ordem **<u>decrescente</u>** os seguintes elementos: **2 8 16 1 3 10 11 12**. Mostre a evolução do espaço de memória para os primeiros 4 elementos. Comece por mostrar no espaço de memória a *Heap Tree* inicial.

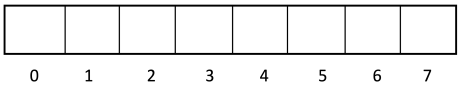


<= *Heap Tree* inicial (resultante do processo de heapenização)



<= c/ 1º elemento ordenado

<= c/ 2º elemento ordenado



<= c/ 3º elemento ordenado

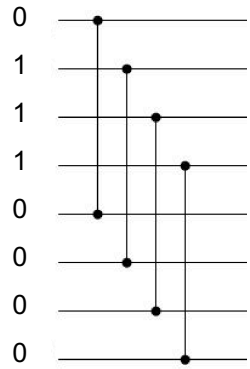
<= c/ 4º elemento ordenado

C.2 (10 pontos) O algoritmo de ordenamento Least Significant Digit Radix Sort (LSD) tem complexidade temporal O(kN), em que assumindo que k é constante podemos dizer que este algoritmo tem complexidade linear. Em alguma circunstância o LSD pode ter complexidade O(N log N)?

_____

Justifique:

C.3 Considere a rede de ordenamento apresentada na figura:



C.3.1 (5 pontos) Mostre na figura acima os valores à saída da rede de ordenamento

C.3.2 (5 pontos) Que tipo de rede é esta?

_____

Caraterize a entrada e as saídas produzidas em termos do que foi estudado sobre redes de ordenamento:

GRUPO D – Mapeamento de Cadeias de Carateres

(este grupo aparecerá em alternativa a outro grupo ou parte de outro grupo)

D.1 (10 pontos) Considere a seguinte cadeia de carateres:

G C T T T A A C T A C G A C A C

e o seguinte padrão a procurar na cadeia acima:

C G A C

Mostre os passos principais do processo de Mapeamento recorrendo ao algoritmo de Boyer-Moore