

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 Época Normal

Duração: 90mn
23 de Junho de 2022

Nome:

N. Est./St ID:

email:

Avaliação para 100 pontos

Área Reservado

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

A.1 (10 pontos) Considere os seguintes blocos de código

Bloco A	Bloco B	Bloco C
<pre>for(int i = 0; i < n; i++) { for(int j = 0; j < n; j=j*2) { } }</pre>	<pre>for(int i = 0; i < n; i++) { for(int j = n-16; j < n; j=j*2) { } }</pre>	<pre>def s(list): i=1 while i != len(list): move_left(list,i) # complexidade O(N) i=i+1 return list</pre>

Qual a complexidade O-Grande do

Bloco A:

Bloco B:

Bloco C:

Justifique sucintamente:

A.2 (10 pontos, cada afirmação incorreta penaliza em 5 pontos) Considere a técnica de desenho de algoritmos Programação Dinâmica. Quais das seguintes afirmações são verdadeiras:

- ☐ necessita de mais tempo de computação ☐ tende a criar uma recursão mais curta (menor profundidade da árvore de recursão) ☐ necessita de memória adicional ☐ é usada para transformar um programa recursivo em iterativo ☐ permite transformar um processo recursivo num iterativo

Justifique sucintamente:

GRUPO B – Estruturas de Dados

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

Prop. #1 _____

Prop. #2 _____

Prop. #3 _____

B.2 (20 pontos) Considere uma árvore VP. Mostre a evolução da árvore inserindo os seguintes elementos:

20; 10; 5; 3; 7; 1; 4; 2;

B.3 (15 pontos) Considere uma Árvore-B em que cada nó vai ocupar um bloco em disco rígido. Cada bloco em disco disponibiliza 4096 bytes, a informação em cada nó ocupa 64 bytes e cada referência para outro bloco 4 bytes. Qual deve ser o fator de ramificação da árvore?

- ☐ 50 ☐ 59 ☐ 32 ☐ 52 ☐ nenhum dos anteriores

Apresente os cálculos que levam à escolha acima.

GRUPO C – Algoritmos de Ordenamento

C.1 (20 pontos) Considere o algoritmo de ordenamento *HeapSort*. 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



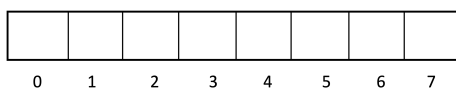
<= c/ 1º elemento ordenado



<= c/ 2º elemento ordenado



<= c/ 3º elemento ordenado

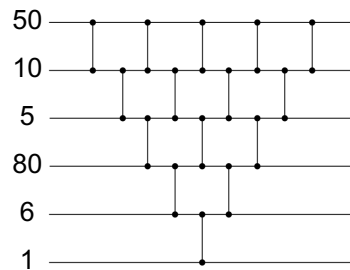


<= c/ 4º elemento ordenado

C2. (10 pontos) Como é sabido 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 de cada comparador bem com os valores à saída da rede de ordenamento

C.3.2 (5 pontos) Que algoritmo de ordenamento é emulada por esta rede?

Justifique:

Folha Rascunho

Folha Rascunho