

**UNIVERSIDADE ESTADUAL DE CAMPINAS**  
Instituto de Computação

Disciplina MC202	Entrega  <b>27/08/2020, 21:00</b>
Professor Iago Augusto de Carvalho	
Monitores Arthur (PAD), Brenner (PED), Deyvison (PED), Enoque (PED), Matteus (PED), Thiago (PAD).	

**Exame final**  
**RA com dígito final par**

### Instruções

Esta prova contém um total de quatro questões. Ela terá início as 19 horas e fim as 21 horas.

A prova deverá ser resolvida a lapis ou a caneta e, posteriormente, fotografada e enviada por email para mim até o prazo limite de 21h20. Desta forma, existe um prazo de 20 minutos para que vocês fotografem a prova e a enviem por e-mail após sua finalização.

O e-mail deverá conter os seguintes dados:

- **Nome completo**
- **RA**
- **Turma** (D, E, F ou G)
- **Fotos da prova em anexo** (em formato .zip ou em uma sequência lógica)

### Questão 1 - Listas encadeadas e vetores (valor: 0.25)

Você tem que implementar uma estrutura de dados para um software específico de um cliente. Você sabe que os dados armazenados nesta estrutura sofrerão diversas operações de inserção e remoção, mas poucas operações de modificação/atualização. Qual estrutura de dados você escolheria para implementar nesta situação? Justifique sua resposta.

**Resposta:** Deve-se utilizar uma lista encadeada. Esta estrutura possui uma baixa complexidade de inserção e remoção, que são as operações mais realizadas na estrutura. Um vetor não deve ser utilizado, pois os custos de remoção são muito altos.

### Questão 2 - Algoritmos de Ordenação (valor: 0.25)

Considere os seguintes algoritmos de ordenação: (i) Insertion sort; (ii) Selection sort; (iii) Quicksort; e (iv) Mergesort. Você recebe dois conjuntos de inteiros  $A$  e  $B$  e deve criar um conjunto  $C = A \cup B$  tal que os elementos em  $C$  estejam ordenados em ordem **crescente**. Considere que  $|A|$  representa o número de elementos contidos em  $A$  e  $|B|$  representa o número de elementos contidos em  $B$ . Nas situações abaixo, qual algoritmo de ordenação você escolheria? Justifique sua resposta.

- (a)  $|A| = 10000$  e  $|B| = 10000$ . Os números em  $A$  e em  $B$  não possuem nenhuma ordenação específica.
- (b)  $|A| = 10000$  e  $|B| = 5$ . Os números em  $A$  estão ordenados em ordem crescente e os números em  $B$  não possuem nenhuma ordem específica.
- (c)  $|A| = 1$  e  $|B| = 10000$ . Os números em  $A$  e em  $B$  estão ordenados em ordem decrescente.

### Respostas:

- (a) Tanto o Quicksort como o Mergesort são adequados, pois a complexidade de ambos os algoritmos são  $O(n \log n)$ . O Mergesort deve ser utilizado caso você não possa tolerar um pior caso de  $O(n^2)$ , enquanto o Quicksort deve ser utilizado caso memória seja um problema.

- (b) O correto seria utilizar o Insertion sort. Este é o algoritmo mais eficiente para inserir um pequeno número de elementos em um vetor já ordenado.
- (c) Quicksort deve ser utilizado devido a fácil localização do pivô inicial.

### Questão 3 - Árvores binárias não balanceadas (valor: 0.25)

Para cada um dos conjuntos de inteiros a seguir, desenhe como será a árvore binária não balanceada obtida ao inserir os números na ordem em que são dados, isto é, da esquerda para a direita.

- (a) 10 12 7 9 8 11 13 15 1 3
- (b) 10 9 8 7 6 4 5 2 3 1
- (c) 10 7 13 11 9 1 3 2 15 7

**Respostas:** Ver Figura 1. A questão (c) será desconsiderada, pois existe um número duplicado (7).

### Questão 4 - Tabelas de hash (valor: 0.25)

Você possui uma tabela de hash de endereçamento aberto implementada como uma lista circular, com tamanho  $n = 10$ . Você deve inserir o conjunto de inteiros 5, 10, 15, 20, 2, 4, 1, 9, 12, 6 nesta tabela hash, sempre da esquerda para a direita (primeiro você insere o número 5, depois o número 10, depois o 15, e assim por diante). Desenhe como será a organização final dos números na tabela para as seguintes funções de hash:

- (a)  $h(x) = 2x \bmod 10$
- (b)  $h(x) = x \bmod 5$
- (c)  $h(x) = (3x - 2) \bmod 10$

**Respostas:** Ver Figura 2.

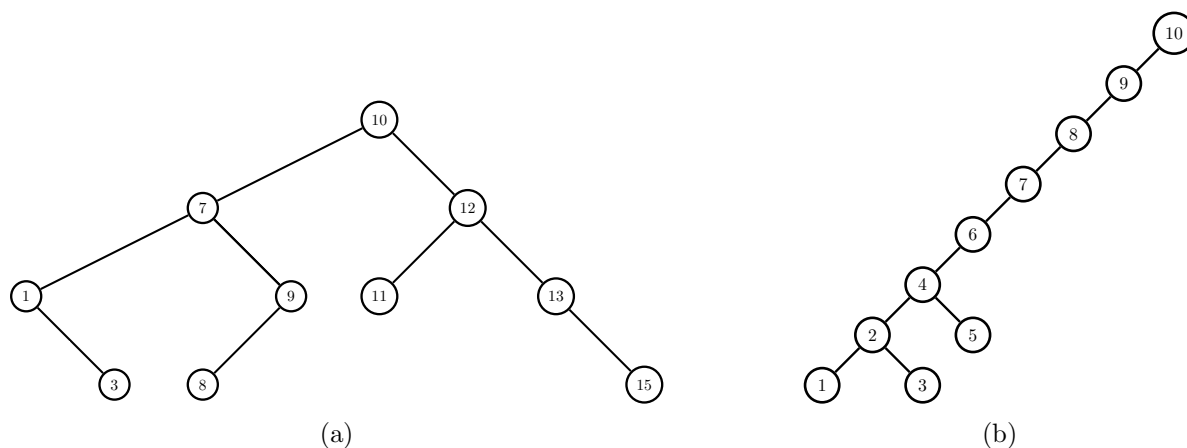


Figura 1: Respostas da questão 3

05	10	15	20	02	01	12	06	04	09
0	1	2	3	4	5	6	7	8	9

(a)

05	10	15	20	02	04	01	09	12	06
0	1	2	3	4	5	6	7	8	9

(b)

04	01	06	05	15	02	09	12	10	20
0	1	2	3	4	5	6	7	8	9

(c)

Figura 2: Respostas da questão 4