

CI-057 : Algoritmos e estruturas de dados III

Prova 2
1º semestre 2019

Instruções para a prova

- A prova é sem consulta;
- A prova dura 1 hora e 40 minutos.

Questão 1 (50 pontos) Marque (V) para verdadeiro e (F) para falso em todas as questões à seguir. Caso marque (F) justifique a resposta. Use a letra correspondente na folha de respostas.

- 1 F a () Uma 3-Heap, ou Heap ternária, é um heap onde cada elemento pode ter 3 filhos. Considere a Heap Ternária com os valores 20, 30, 31, 21, 31, 32, 33, 34, 22, 20, 23. A heap mantém a propriedade de **Heap Mínima**.
- F b () Dada a tabela de frequência das seguintes letras: a - 25, b - 20, s - 40, i - 5, e - 10. A palavra de entrada **saaiib** será codificada como 0101011001100101.
- 7 c () Em uma tabela Hash com endereçamento aberto com sondagem linear, o agrupamento primário pode ser resolvido com uma alteração das constantes na mesma função linear aplicada originalmente na inclusão de elementos.
- V d () Dada chaves **CASA, CARRO, DADO e DIA**. A árvore Patricia resultante terá 7 nós.
- F e () Dada a função hash $h(k) = k \% m$: a inclusão das chaves 7, 26, 19, 21, 46, 18, 30, e 10 em uma tabela hash com $m = 7$ posições não gera colisões.

Questão 2 (25 pontos) Uma 3-Heap, ou Heap ternária, é um heap onde cada elemento pode ter 3 filhos. A representação em vetor segue uma estrutura semelhante aquela usada no heap binário. Na 3-heap, se a raiz da estrutura ocupa a posição 1 do vetor, as próximas 3 posições, de 2 até 4, são ocupadas pelos elementos que são filhos da raiz. A mesma lógica segue recursivamente para os filhos. A função $filho(i, n)$ retorna o índice do n -ésimo filho do nó de índice i . Por exemplo, $filho(1, 1)$ retorna 2, $filho(1, 3)$ retorna 4, $filho(3, 2)$ retorna 9. Escreva um algoritmo em C ou em pseudocódigo semelhante a C que implemente a função $filho(i, n)$ para uma 3-heap.

Questão 3 : (25 pontos)

Implemente uma estrutura de dados e código em C ou em pseudocódigo semelhante a C para decodificar um conjunto de caracteres de uma árvore de Huffman.

