Aluno(a):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ matrícula:\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| 1a) | 1.0 |  |
| 2a) | 1.0 |  |
| 3a) | 3.0 |  |
| 4a) | 2.0 |  |
| 5a) | 3.0 |  |
|  |  |  |

1. A prova é individual e sem consulta.
2. A interpretação faz parte da questão.
3. O tempo de prova é 1:30 h.
4. As respostas devem seguir as questões.
5. Caso precise de rascunho use o verso da folha.
6. Caso parte da resposta esteja no verso, indique claramente este fato.
7. (1.0 ponto) Faça a representação binária da seguinte árvore n-ária.



J

Resp.:

Uma modificação direta resultaria em:



J

Rotacionando a estrutura para visualização teríamos:



J

1. (1.0 ponto) Desenhe a árvore binária a partir dos seus percursos em pós-ordem e in-ordem (ordem simétrica). Indique a pré-ordem para a árvore obtida.

Pós-ordem: Y B T P A X V L R H K J

In-ordem: Y T B X P A J L V K R H

Resp.:

|  |
| --- |
|  |

1) Y B T P A X V L R H K J indica que J é a raiz;

Logo, Y T B X P A J L V K R H indica que Y T B X P A é a SAE de J e L V K R H é a SAD de J

J

Y T B X P A L V K R H

2) Recursivamente, Y B T P A X indica que X é a raiz da SAE

Logo, Y T B X P A indica que Y T B é a SAE de X e P A é a SAD de X

e assim por diante.

A pré-ordem será: J X T Y B A P K L V H R

1. (3.0 pontos) Escreva em C uma função para determinar se uma árvore é AVL. A função deve receber como parâmetro o endereço do nó raiz da árvore e retornar o valor 1 se for AVL e 0 caso contrário. Use a seguinte estrutura:

|  |
| --- |
| **struct** no {  **void** \*chave;  **struct** no \*esq, \*dir;  }; |

Resp.:

#include <stdlib.h>

struct no {

void \*chave;

struct no \*esq, \*dir;

};

int checkAVL( no \*node )

if( node==NULL ) return 0; /\* base da recursão: folha tem altura 0

int esq = checkAVL( node->esq ); /\* testa recursivamente SAE

int dir = checkAVL( node->dir ); /\* testa recursivamente SAD

if( ( dir < 0 ) || ( esq < 0 ) ) return -1; /\* se a SAE ou a SAD não é AVL, repasse -1

if( abs( dir - esq ) > 1 ) return -1; /\* se detectou desbalanceamento, sinalize com -1

return( max( abs( dir ), abs( esq ) ) + 1 ); /\* calcule a altura \*/

}

int AVL( no \*node ) {

if( checkAVL( node ) >= 0 ) return 1;

else return 0;

}

1. (2.0 pontos)

a. (1.0 ponto) Qual é o número máximo de chaves que uma árvore 2-3 e altura *h* pode armazenar (uma árvore que só possui a raiz tem altura 0)? Explique sua resposta.

Dica: A soma dos *n* primeiros termos de uma série geométrica é dada por:

*a + ar + ar2 + ar3+ ...+ arn-1 = a*

b. (1.0 ponto) Qual será a menor altura de uma árvore 2-3 que armazena 80 chaves? Explique sua resposta.

Resp.:

a) Uma árvore 2-3 pode ter no máximo *N* chaves, onde

*N = (m-1)\*(1 + m + m2 + m3+ ... + mh) = (m-1) = mh+1 - 1*  onde *m*=3

a) Uma árvore 2-3 com 80 chaves deverá ter altura *h*, onde *h* é o menor inteiro tal que:

80 ≤ 3*h+1 – 1*

81 ≤ 3*h+1*

Logo, *h =3*

1. (3.0 pontos)

a. (1.5 ponto) Crie uma árvore 2-3 através da inserção das seguintes chaves, exatamente na

ordem apresentada: 1, 3, 6, 8, 2, 32.

Mostre todos os passos.

b. (1.5 ponto) Remova a chave 8 da árvore resultante de (a).

Mostre e explique todos os passos.

Resp.:

(a) Inserções de 1, 3, 6, 8, 2, 32:

|  |  |
| --- | --- |
| 1 |  |

|  |  |
| --- | --- |
| 1 | 3 |

|  |  |  |
| --- | --- | --- |
| 1 | 3 | 6 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 3 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 |  |  | 6 |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 3 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 |  |  | 6 | 8 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 3 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 |  | 6 | 8 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 3 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 |  | 6 | 8 | 32 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 3 | 8 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 |  | 6 |  |  | 32 |  |

(b) Remoção de 8:

Árvore inicial:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 3 | 8 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 |  | 6 |  |  | 32 |  |

1) Troque 8 com a sua sucessora, 32

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 3 | 32 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 |  | 6 |  |  | 8 |  |

2) Remova 8

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 3 | 32 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 |  | 6 |  |  |  |  |

3) Elimine o nó vazio, movendo a maior chave do pai para o filho do meio.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 3 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 |  | 6 | 32 |

Ou, alternativamente:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 2 | 6 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  | 3 |  |  | 32 |  |