Raissa Pereira Miranda CJ3028941

Prova Teórica de Revisão – Estrutura de Dados

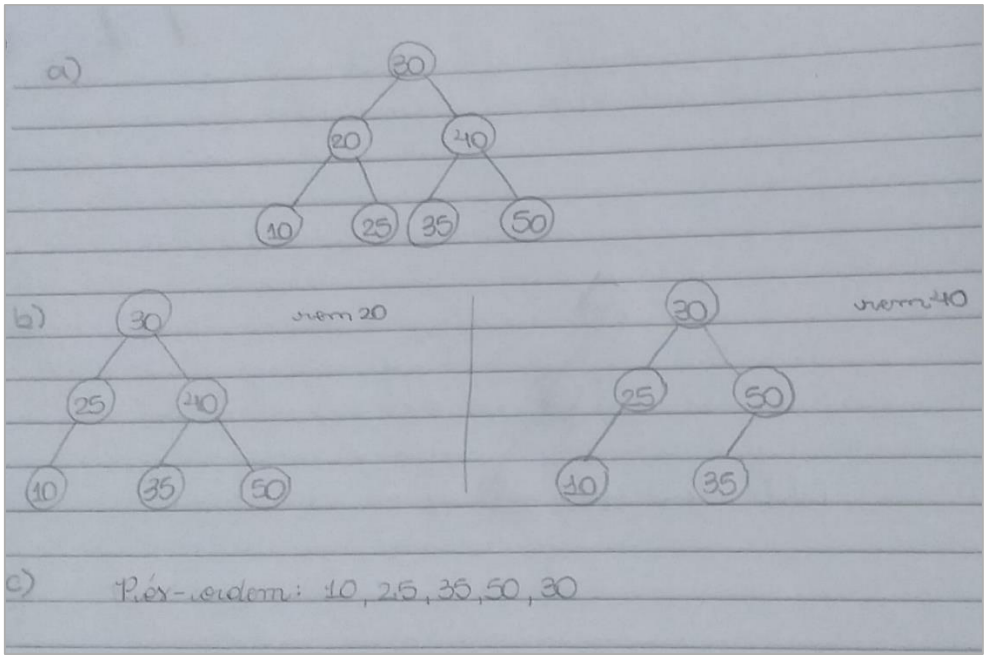
Questão 1 – Árvore Binária de Busca (ABB): Inserção e Remoção

Considere a seguinte sequência de inserções em uma árvore binária de busca vazia: 30, 20, 40, 10, 25, 35, 50

a) Desenhe a árvore resultante após todas as inserções.

b) Remova os nós 20 e 40. Apresente o novo desenho da árvore após as remoções.

c) Qual é o percurso pós ordem da árvore final?



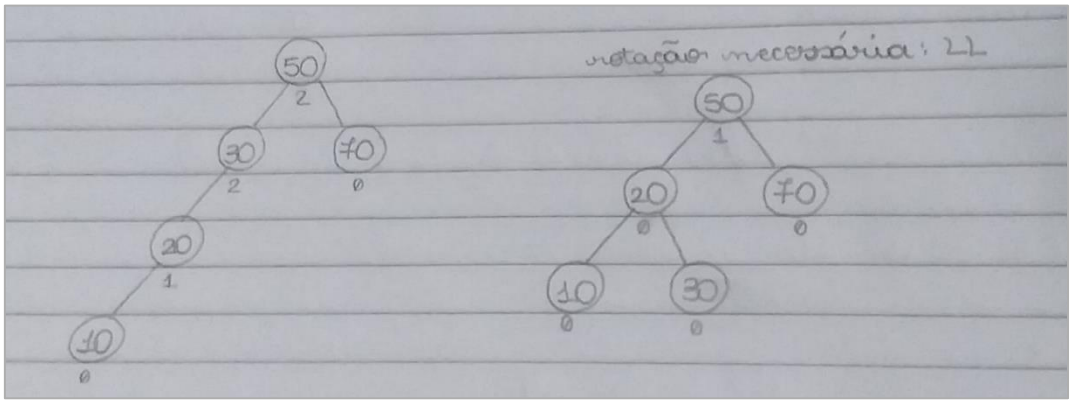
Questão 2 – Árvores AVL

Considere a inserção sequencial dos seguintes elementos em uma árvore AVL: 50, 30, 70, 20, 10

a) Aponte em que momento ocorre o desequilíbrio e que tipo de rotação é necessário para manter a árvore balanceada.

R: O desequilíbrio ocorre quando a chave 10 é inserida, sendo necessário realizar a rotação LL para balancear a árvore.

b) Esboce a árvore final após a rotação.



Questão 3 – Árvores N-árias e Tries

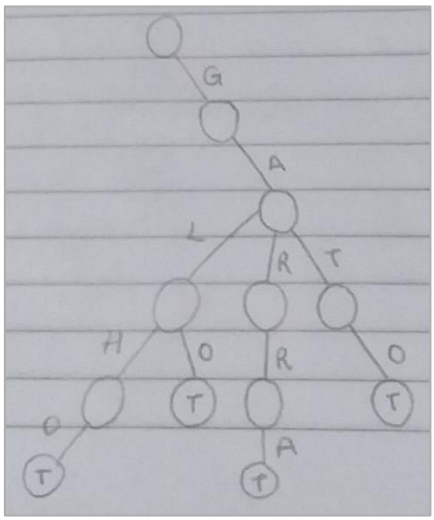
a) Explique a principal diferença entre uma árvore N-ária e uma Trie.

R: A árvore n-ária armazena qualquer tipo de estrutura hierárquica, guardando explicitamente as chaves completas. Já a árvore trie é projetada pra armazenar palavras com base em seus prefixos, guardando um caractere ou parte da chave.

b) Dê um exemplo prático de aplicação de Tries.

R: Um exemplo de aplicação prática de Tries seria na implementação de um corretor ortográfico, em que elas armazenam um dicionário de palavras para verificar se a palavra digitada existe e sugerir correções.

c) Dado o conjunto de palavras { "gato", "galho", "galo", "garra" }, desenhe uma Trie representando todas elas.



Questão 4 – Pilhas e Filas

a) Simule a execução de uma pilha, considerando as seguintes operações:

push(1), push(2), push(3), pop(), push(4), pop()

Qual é o conteúdo final da pilha (do topo para a base)?

R: O conteúdo final do topo para a base seria: 2, 1.

b) Simule a execução de uma fila, considerando as seguintes operações:

enqueue(5), enqueue(6), dequeue(), enqueue(7), dequeue()

Qual é o conteúdo final da fila (do início para o fim)?

R: O conteúdo final do início para o fim seria: 7.

Questão 5 – Listas Encadeadas

Explique a diferença entre:

a) Lista Simplesmente Encadeada

b) Lista Duplamente Encadeada

c) Lista Circular

Para cada uma, mencione uma vantagem ou aplicação típica.

R: Na lista simplesmente encadeada cada nó possui apenas um ponteiro indicando o próximo nó da lista. Uma aplicação típica dela seria na implementação de estruturas lineares, como pilhas e filas. Uma vantagem seria usar menos memória do que listas duplamente encadeadas.

Na lista duplamente encadeada cada nó possui ponteiros tanto para o próximo nó quanto para o nó anterior da lista. Uma aplicação típica seria o botão de próximo e anterior do navegador. Uma vantagem seria que é possível percorrer a lista em ambas as direções.

Na lista circular o último nó aponta de volta para o primeiro nó. Uma aplicação típica seria na implementação de escalonadores de tarefas, como o Round-Robin. Uma vantagem seria que por ser circular, não há necessidade de reiniciar a iteração manualmente.

Questão 6 – Recursividade

Considere a seguinte função recursiva:

```
int f(int n) {
if (n == 0) return 0;
return n + f(n - 1);
}
```

a) Qual é o valor retornado por f(4)?

R: O valor retornado é 10.

f(4) = 4 + f(3)

f(3) = 3 + f(2)

f(2) = 2 + f(1)

f(1) = 1 + f(0)

f(0) = 0

b) Descreva o que essa função calcula.

R: Essa função calcula o somatório de N usando recursividade.

c) Essa função pode ser reescrita de forma iterativa? Justifique.

R: Pode ser reescrita de forma iterativa, pois, por meio de um laço 'for', é possível realizar a soma de 0 até N. Para isso, se inicializa uma variável 'soma' com zero e, em seguida, vai acumulando nela os valores de cada iteração. Ao finalizar o laço, é retornado o resultado do somatório.

Questão 7 – Arrays vs Listas

Compare arrays e listas encadeadas com relação a:

a) Acesso a elementos

R: Os arrays possem um acesso melhor por ser feito direto pelo índice, tendo um tempo constate de O(1). Já nas listas encadeadas o acesso é feito sequencialmente, tendo um tempo linear de O(n).

b) Inserção no início

R: Os arrays seriam menos eficientes para a inserção no início, visto que seria preciso deslocar todos os elementos para alcançar essa posição. Já nas listas encadeadas seria só criar um novo nó e apontar ele para o antigo início.

c) Uso de memória

R: Os arrays, quando bem implementados, podem ser melhores em uso de memória, pois armazenam apenas os dados lado a lado. Já nas listas encadeadas é utilizado mais memória, pois, além dos dados, cada nó armazena também ponteiros.

Dê exemplos práticos de onde cada um seria mais indicado.

R: Os arrays seriam mais indicados em aplicações que precisam de fácil acesso ao índice como em vetores de notas de alunos ou em jogos que utilizam estruturas fixas, como o Sudoku. Por outro lado, as listas encadeadas seriam melhores na implementação de pilhas e filas que podem precisar de alterações nos extremos.