

Matéria: Programação

Assunto: Árvores Binárias

Resumo Teórico do Assunto

Para dominar questões sobre Árvores Binárias, especialmente aquelas que envolvem a reconstrução ou identificação de árvores a partir de sequências de nós, é fundamental compreender a estrutura da árvore e, principalmente, os diferentes tipos de **percurso (traversal)**.

Árvores Binárias: Conceitos Essenciais e Percurso

Uma **Árvore Binária** é uma estrutura de dados hierárquica onde cada **nó** pode ter no máximo dois **filhos**: um filho à esquerda e um filho à direita. Elas são amplamente utilizadas para organizar dados de forma eficiente, permitindo operações como busca, inserção e remoção.

1. Terminologia Fundamental

- **Nó (Node)**: É o elemento básico da árvore. Cada nó contém um valor (dado) e referências (ponteiros) para seus filhos esquerdo e direito.
- **Raiz (Root)**: É o nó superior da árvore. Uma árvore tem apenas uma raiz, e ela não possui um nó pai.
- **Filho (Child)**: Um nó que é descendente direto de outro nó. Um nó pode ter um filho esquerdo e/ou um filho direito.
- **Pai (Parent)**: Um nó que possui um ou mais filhos.
- **Folha (Leaf)**: Um nó que não possui nenhum filho. São os nós terminais da árvore.
- **Subárvore (Subtree)**: Uma árvore formada por um nó e todos os seus descendentes. Cada filho de um nó raiz é a raiz de uma subárvore.

2. Percurso (Traversal) em Árvores Binárias

O percurso em uma árvore é o processo de visitar cada nó da árvore exatamente uma vez, seguindo uma ordem específica. Existem três métodos principais de percurso recursivo, cada um com uma ordem de visitação diferente:

1. Percurso em Pré-Ordem (Pre-order Traversal):

* **Ordem de Visitação**: Raiz → Subárvore Esquerda → Subárvore Direita

* **Explicação**: Primeiro, visita-se o nó atual (a raiz da subárvore em questão), depois percorre-se recursivamente toda a subárvore esquerda, e por fim, percorre-se

recursivamente toda a subárvore direita.

* **Aplicação Comum:** Útil para criar uma cópia da árvore ou para representar expressões em notação prefixa.

2. Percurso Em Ordem (In-order Traversal):

* **Ordem de Visitação:** Subárvore Esquerda → Raiz → Subárvore Direita

* **Explicação:** Primeiro, percorre-se recursivamente toda a subárvore esquerda, depois visita-se o nó atual (a raiz da subárvore), e por fim, percorre-se recursivamente toda a subárvore direita.

* **Aplicação Comum:** Para Árvores de Busca Binária (BSTs), este percurso resulta em uma sequência ordenada dos valores dos nós.

3. Percurso em Pós-Ordem (Post-order Traversal):

* **Ordem de Visitação:** Subárvore Esquerda → Subárvore Direita → Raiz

* **Explicação:** Primeiro, percorre-se recursivamente toda a subárvore esquerda, depois percorre-se recursivamente toda a subárvore direita, e por fim, visita-se o nó atual (a raiz da subárvore).

* **Aplicação Comum:** Útil para deletar a árvore (liberar memória de baixo para cima) ou para representar expressões em notação pós-fixa.

3. Conhecimento Essencial para Resolver as Questões

Para resolver a questão apresentada, que pede para identificar a árvore a partir de uma sequência de percurso em pré-ordem, você precisa:

- **Dominar a regra do Percurso em Pré-Ordem:** Lembre-se sempre: **Raiz, Esquerda, Direita**.
- **Identificar a Raiz:** Em um percurso em pré-ordem, o **primeiro elemento da sequência é sempre a raiz** da árvore (ou da subárvore que está sendo percorrida).
- **Aplicar a Recursividade:** Após identificar a raiz, os elementos subsequentes na sequência corresponderão aos nós da subárvore esquerda e, em seguida, da subárvore direita. A dificuldade reside em determinar onde a subárvore esquerda termina e a subárvore direita começa, o que geralmente exige a análise das opções de árvore fornecidas ou a combinação com outro tipo de percurso (como o em ordem) para reconstruir a árvore de forma única.
- **Testar as Opções:** Dada uma sequência de pré-ordem e múltiplas opções de árvores, a estratégia mais eficaz é aplicar o percurso em pré-ordem a cada uma das árvores candidatas e verificar qual delas gera a sequência fornecida.

Ao aplicar a regra **Raiz, Esquerda, Direita** de forma sistemática e recursiva, você será capaz de traçar o percurso de qualquer árvore binária e, inversamente, identificar a árvore correta a partir de sua sequência de percurso.

Questões de Provas Anteriores

Fonte: escrituario_agente_de_tecnologia.pdf, Página: 22

pcimarkpci MjgwNDowMTRkOjE0YTU6OTI1ODozOGQ2OjNhMGM6NTM0MzplZmI1:U3V
uLCAyNyBKdWwgMjAyNSAyMzo0NzozMSAtMDMwMA==
www.pciconcursos.com.br

22

BANCO DO BRASIL

AGENTE DE TECNOLOGIA - Microrregião 158 -TI GABARITO 1

67

Um estudante de computação decidiu escrever um método Java para exibir, no console, em pré-ordem, os valores dos nós de uma árvore binária recebida como parâmetro. Ao executar esse método, os seguintes valores foram exibidos no console:

80 84 55 76 72

Considerando os valores exibidos, qual árvore foi recebida como parâmetro?

(A)

(B)

(C)