

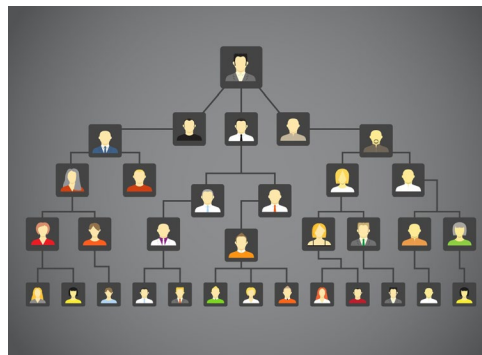
Árvores

Árvores Binárias

Prof. Leandro Colevati

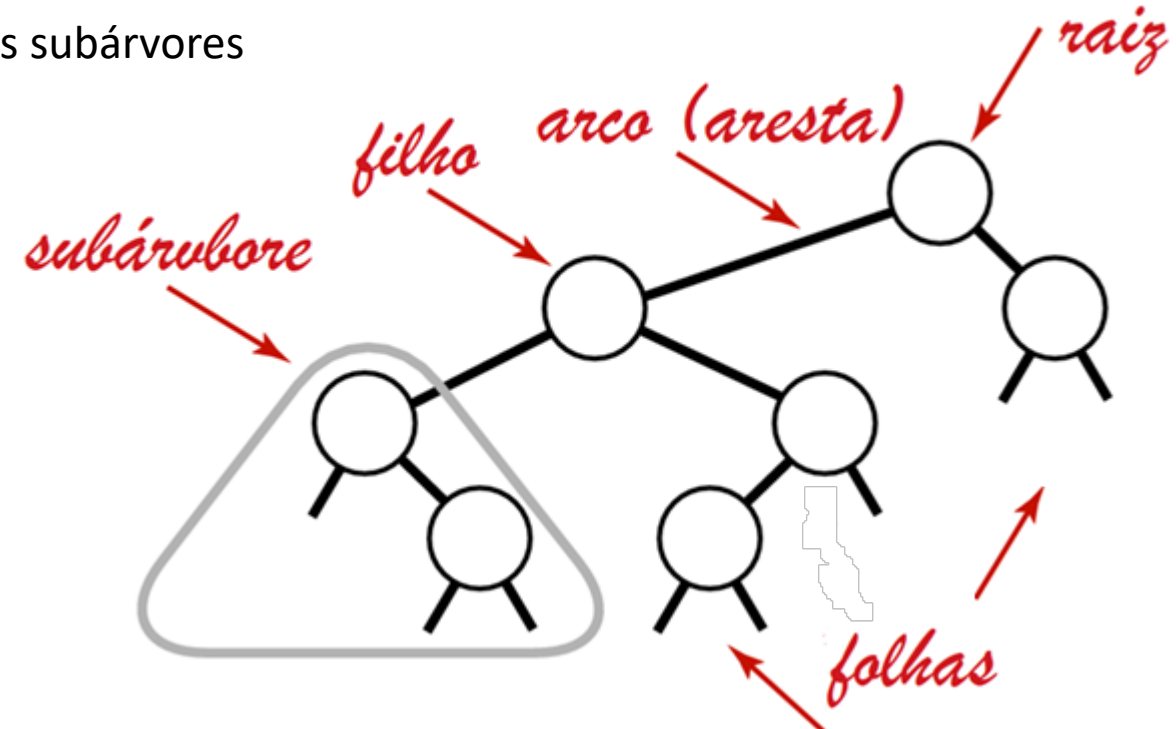
Introdução

- Estruturas estudadas até agora não são adequadas para representar dados que devem ser dispostos de maneira hierárquica
- Exemplos:
 - Hierarquia de pastas (Sistemas de Arquivos)
 - Árvore genealógica
- Árvores são estruturas adequadas para representação de hierarquias
- O conceito de árvores está diretamente ligado com recursividade



Introdução

- Elementos básicos:
 - Grau
 - Número de subárvores de um nó
 - Raiz
 - Primeiro elemento da árvore, com zero ou mais subárvores
 - Folha
 - Nós sem filhos
 - Filhos
 - Nós raízes de uma subárvore
 - Arco (Aresta)
 - Conexão entre dois nós

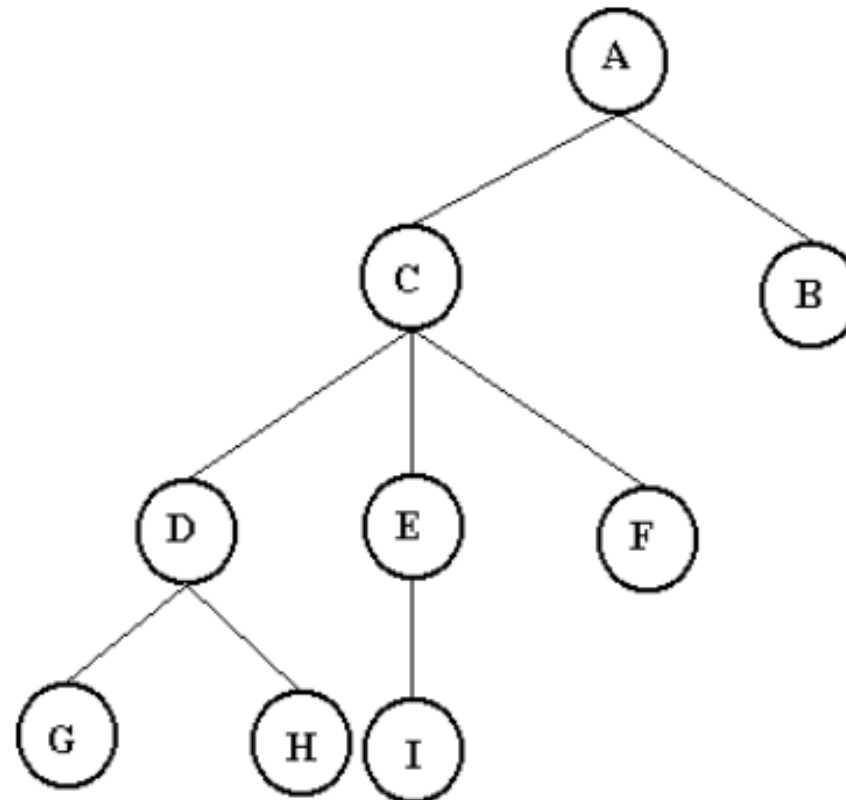


Introdução

- Propriedades:
 - Cada vértice (exceto a raiz) tem exatamente um antecessor (pai)
 - Cada vértice tem nós sucessores imediatos, com exceção das folhas (ou terminais)
 - Filhos de um mesmo pai → irmãos
 - Nós com pelo menos um filho → Internos (ou Não-terminais)
 - Caminho é uma lista de vértices distintos e sucessivos conectados por arcos
 - Existe exatamente um caminho entre a raiz e cada um dos nós da árvore
 - Qualquer nó é a raiz de uma subárvore consistindo dele e dos nós abaixo

Grau

- Grau de um vértice:
 - Número de subárvores não vazias de um nó
- Exemplo:
 - Grau de A = 2
 - Grau de C = 3
 - Grau de E = 1

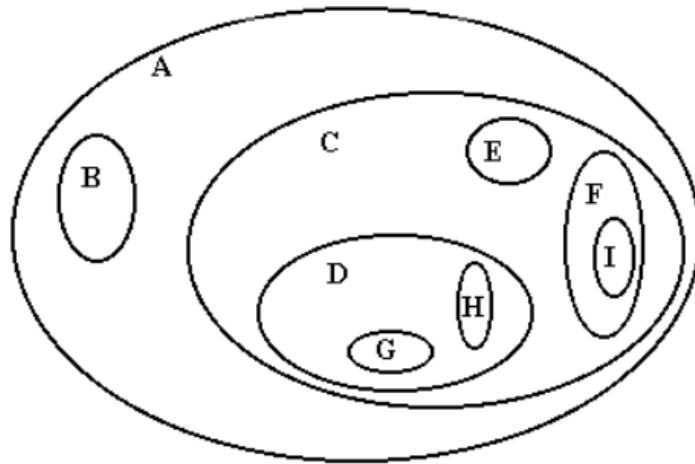


Representação

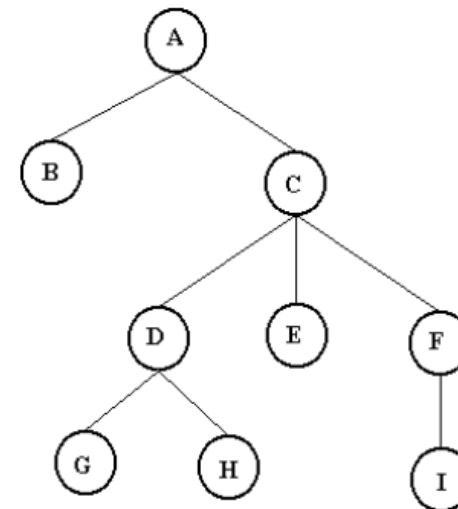
Parênteses aninhados:

– (A (B) (C (D (G) (H)) (E) (F (I))))

Diagrama de Venn (ou Diagrama de Inclusão):

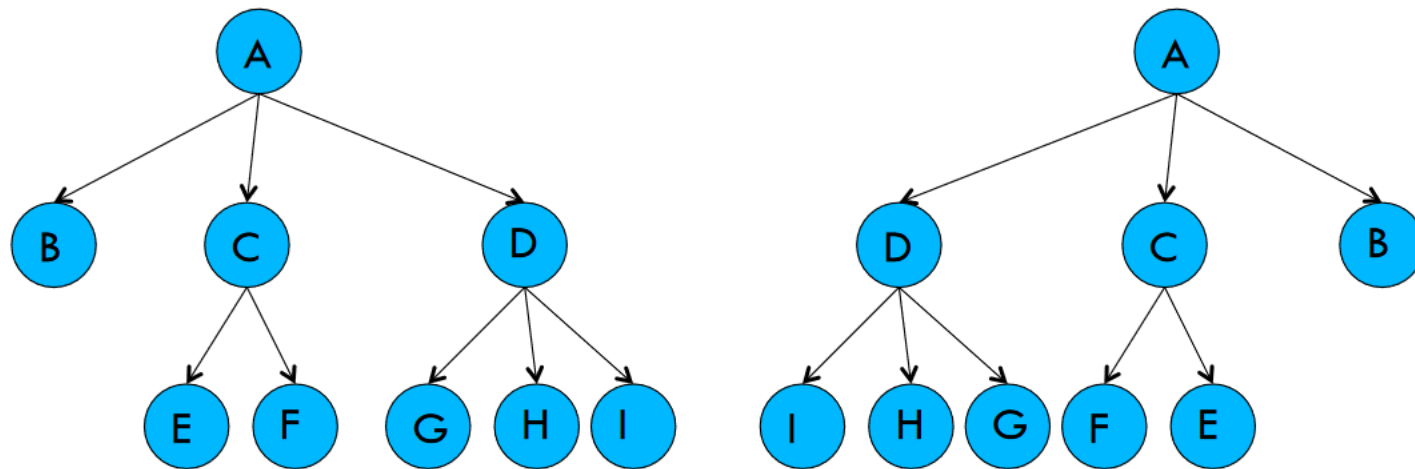


Representação Hierárquica:



Ordenação

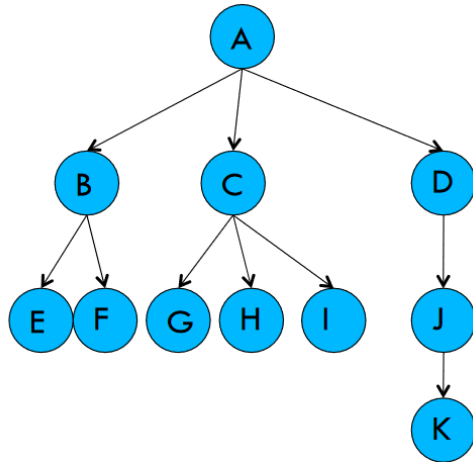
- Considerando as duas árvores abaixo:



- A única diferença entre elas é a ordem das subárvores
 - Uma árvore ordenada é definida com uma árvore onde as subárvores formam um conjunto ordenado
 - Em uma árvore ordenada define-se o primeiro, segundo, até o último irmão, de acordo com alguma propriedade

Vértices

- Os vértices da árvores estão classificados em níveis:
 - Nível é o número de nós entre o vértice e a raiz



Nível da raiz → 0

Nível de C → 1

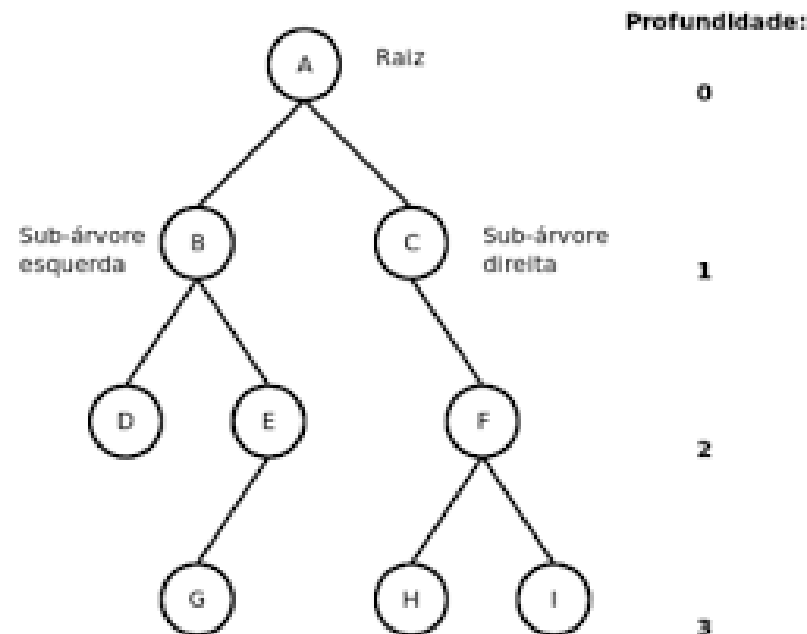
Nível de K → 3

Nível de um nó qualquer → Nível de seu pai + 1

- A altura de uma árvore corresponde à maior distância entre a raiz e o maior nível
 - Árvore nula tem altura 0.
- Floresta é um conjunto de árvores
 - Se removermos a raiz e os arcos que a ligam às subárvores, temos uma floresta

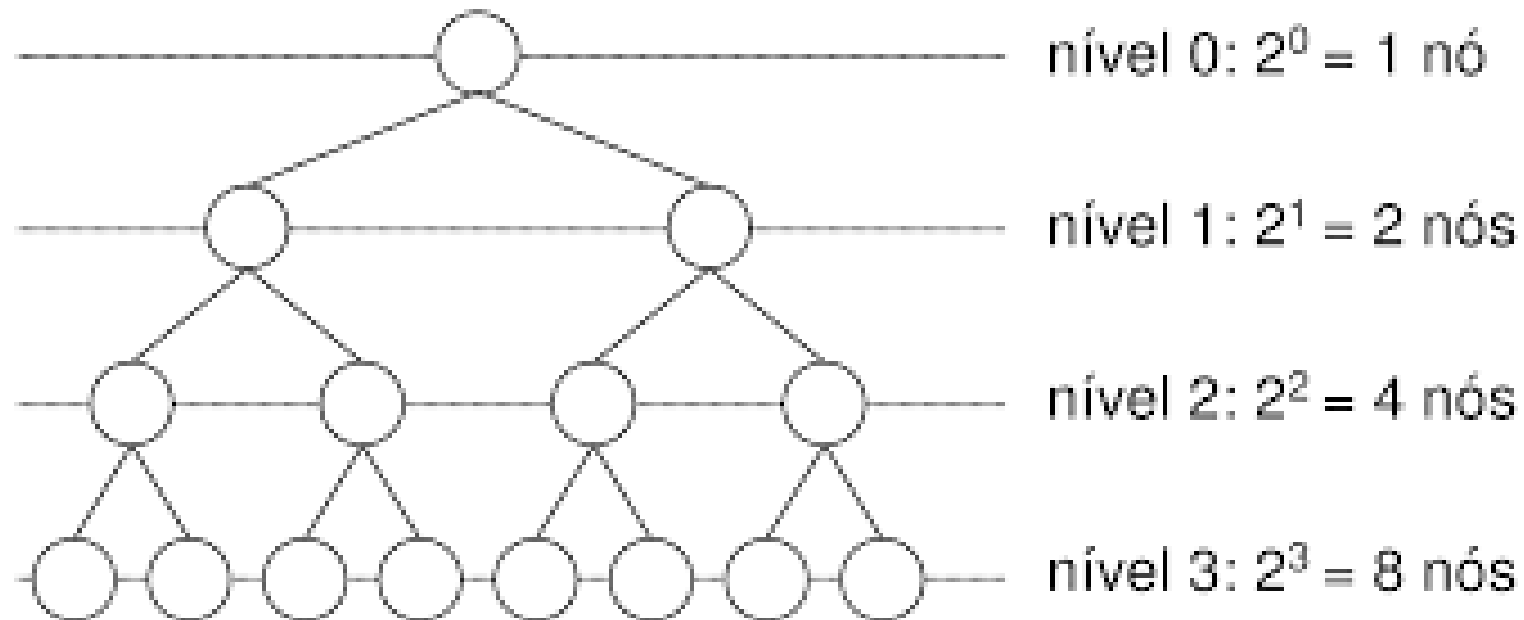
Árvores Binárias

- Árvore nula ou com as seguintes características:
 - Existe um nó especial denominado raiz
 - Nenhum nó tem grau superior a 2, ou seja, nenhum nó tem mais de 2 filhos
 - Existe um senso de posição, ou seja, distingue-se subárvore esquerda e subárvore direita



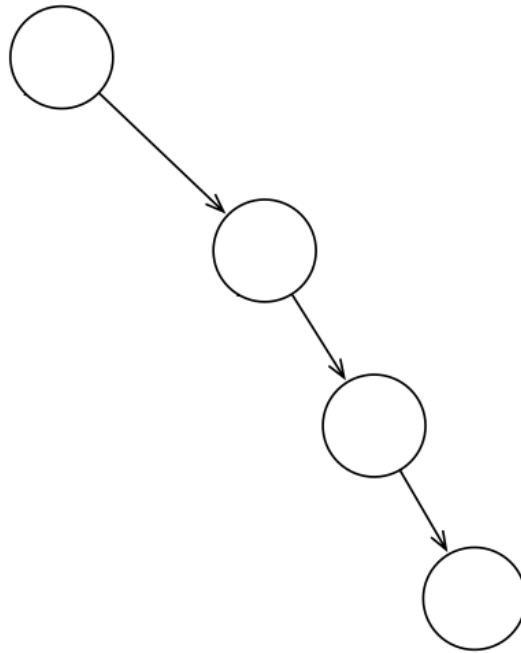
Árvores Binárias

- Considera-se uma árvore binária cheia quando todos os nós (com exceção do último nível) possuem exatamente 2 nós



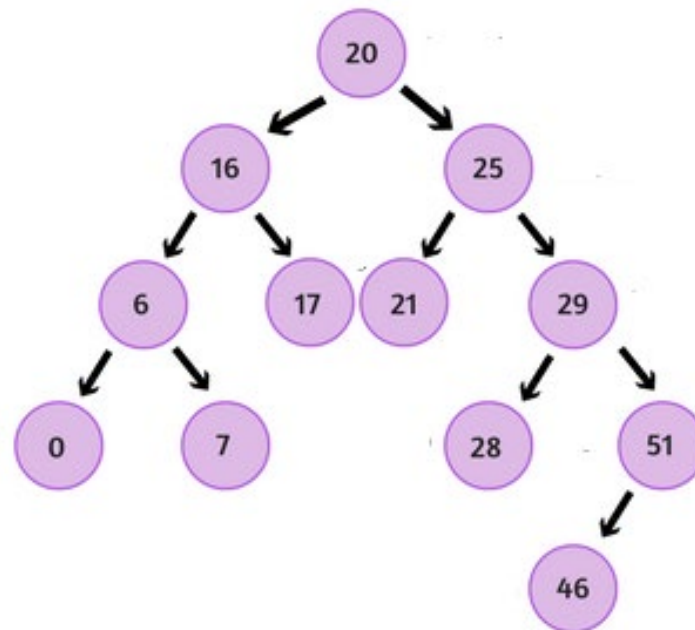
Árvores Binárias

- Uma árvore binária é dita degenerada quando cada nó tem apenas 1 filho e o número de níveis é igual ao número de nós



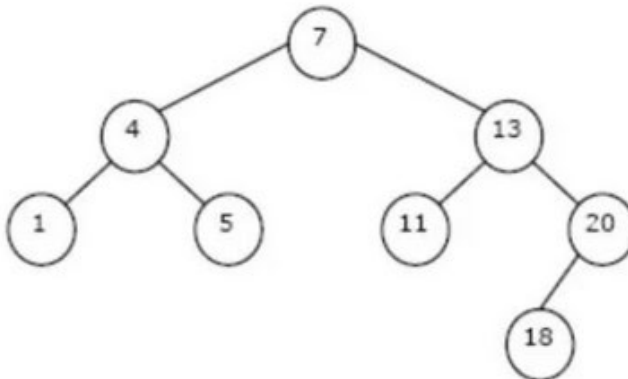
Árvores de Busca Binária

- Uma árvore é denominada Árvore de Busca Binária se:
 - **Todo** elemento da subárvore esquerda é menor que o elemento raiz
 - **Nenhum** elemento da subárvore direita é menor que o elemento raiz
 - As subárvores esquerda e direita também são de busca binária



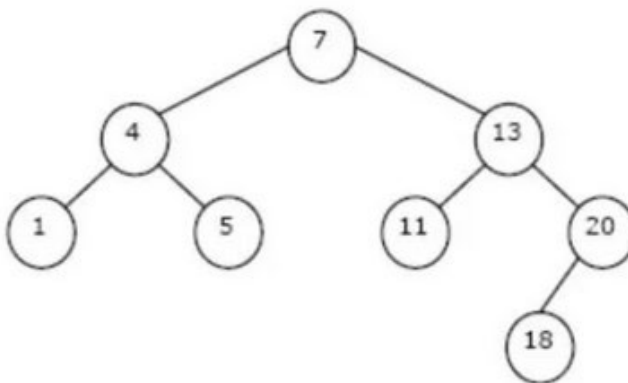
Árvores de Busca Binária

- Inserção
 - Se a árvore está vazia, cria um novo nó e insere as informações do novo nó
 - Se não está, compara a chave a ser inserida com a chave do nó analisado
 - Se menor, insere o valor na subárvore à esquerda
 - Se maior, insere o valor na subárvore à direita
 - Se igual, já existe
- Exemplo:
 - Criar uma árvore de busca binária com os elementos : (7, 13, 20, 4, 1, 18, 5, 11)



Árvores de Busca Binária

- Busca
 - Se o valor procurado for igual à raiz, o valor existe na árvore
 - Se não for, verificar:
 - Se menor, busca o valor na subárvore à esquerda, e assim, recursivamente, em todos os nós da subárvore
 - Se maior, busca o valor na subárvore à direita, e assim, recursivamente, em todos os nós da subárvore
 - A busca cessa se o valor for encontrado

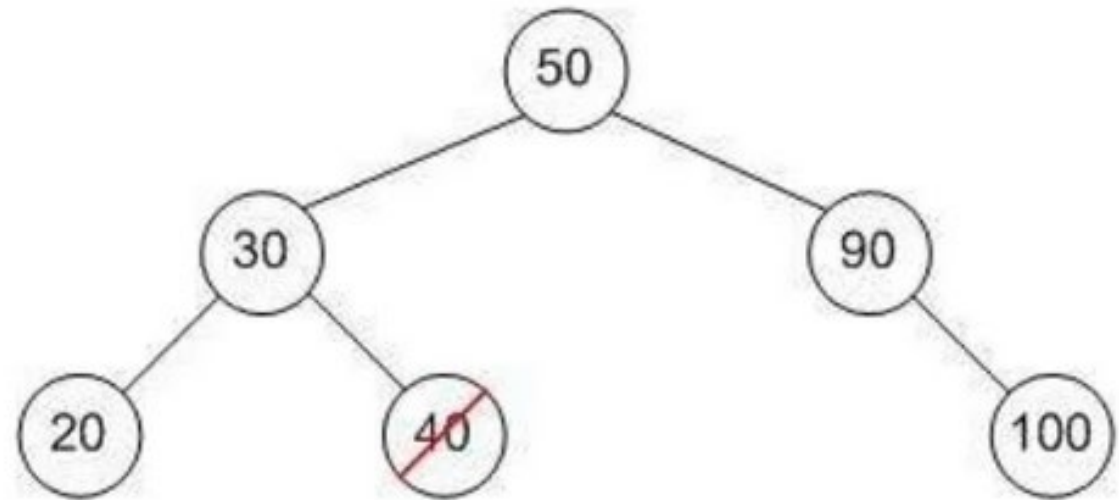


Árvores de Busca Binária

- Remoção
 - Processo complexo (Simplificá-lo pode implicar em perda massiva de dados)
 - Três possibilidades:
 - Remoção na folha
 - Remoção de um nó com 1 filho
 - Remoção de um nó com 2 filhos

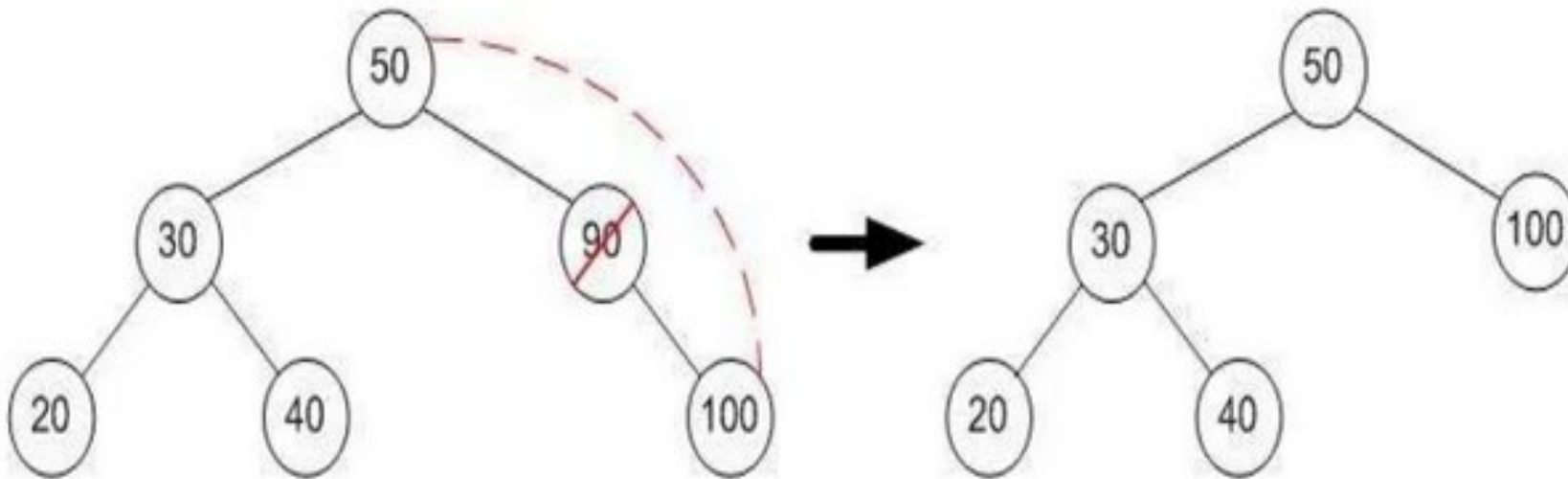
Árvores de Busca Binária

- Remoção na folha
 - Basta remover a folha da árvore
 - Se não for folha, todas as subárvore dela serão removidas junto
 - Exemplo:
 - Sistemas de Arquivos
 - Bancos de Dados Hierárquicos



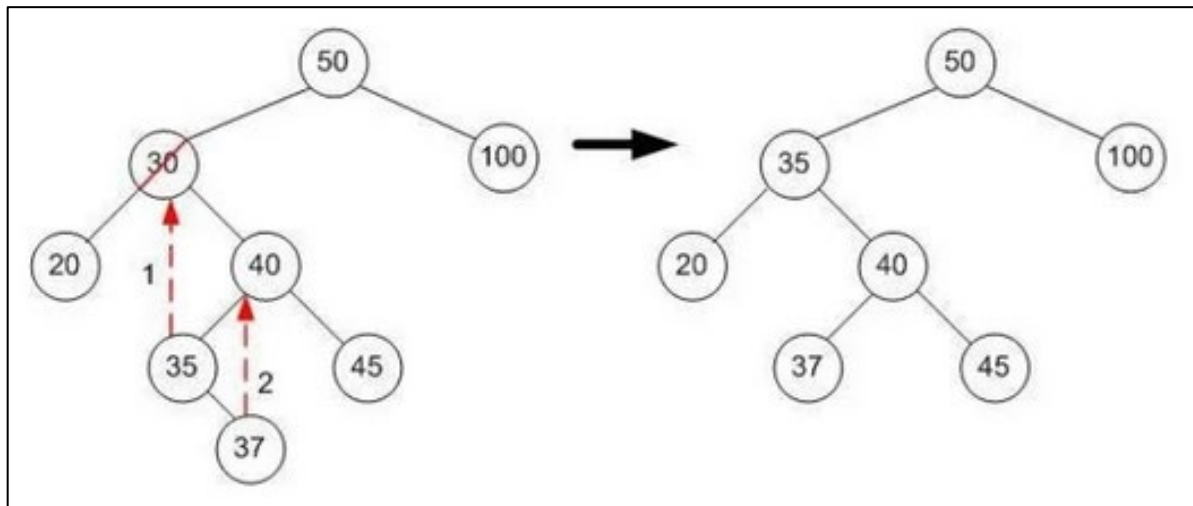
Árvores de Busca Binária

- Remoção de um nó com um filho
 - O filho sobe para a posição do pai

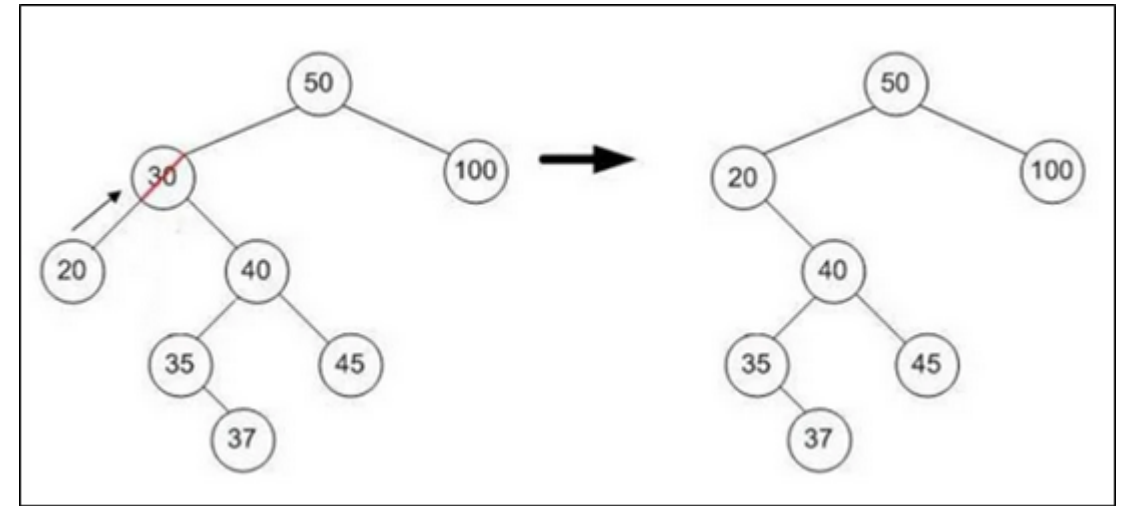


Árvores de Busca Binária

- Remoção de um nó com dois filhos
 - Uma das duas estratégias pode ser adotada:
 - a) Substitui-se o nó pelo nó mais à esquerda da subárvore direita
 - b) Substitui-se o nó pelo nó mais à direita da subárvore esquerda



a)



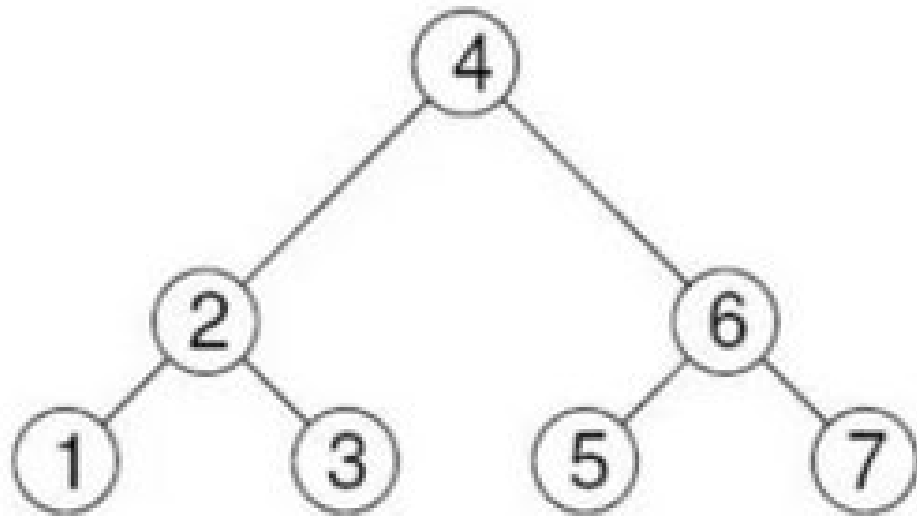
b)

Árvores de Busca Binária

- Atravessamento ou caminhamento da árvore é a passagem, de forma sistemática, por um de seus nós:
- Existem formas diferentes de percorrer os nós de uma árvore:
 - Pré-ordem ou prefixa (Busca em profundidade)
 - Em ordem ou infixada (Ordem central)
 - Pós-ordem ou posfixa

Árvores de Busca Binária

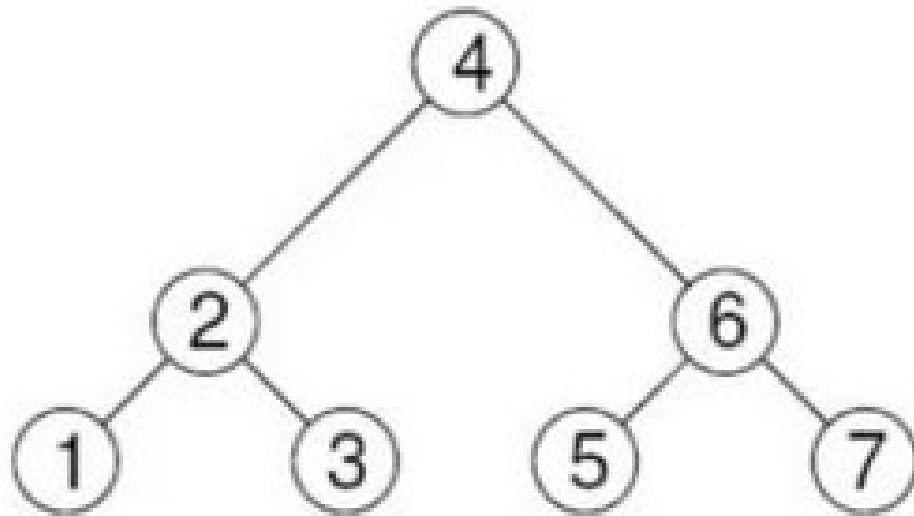
- Atravessamento em pré-ordem (prefixa):
 - Visitar a raiz
 - Caminhar, recursivamente, na subárvore à esquerda
 - Caminhar, recursivamente, na subárvore à direita



4, 2, 1, 3, 6, 5, 7

Árvores de Busca Binária

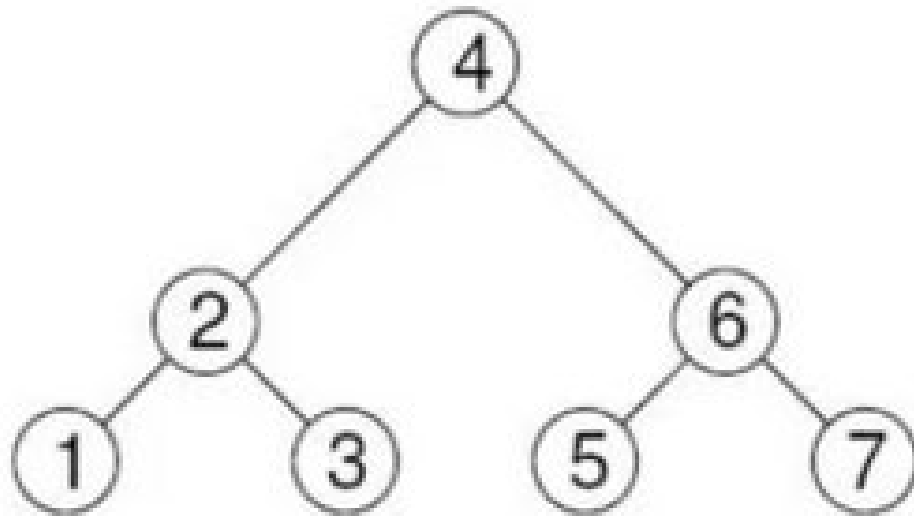
- Atravessamento em ordem (infixa):
 - Caminhar, recursivamente, na subárvore à esquerda
 - Visitar a raiz
 - Caminhar, recursivamente, na subárvore à direita



1, 2, 3, 4, 5, 6, 7

Árvores de Busca Binária

- Atravessamento em pós-ordem (posfixa):
 - Caminhar, recursivamente, na subárvore à esquerda
 - Caminhar, recursivamente, na subárvore à direita
 - Visitar a raiz



1, 3, 2, 5, 7, 6, 4