

Algoritmos e Estruturas de Dados II

Árvores de pesquisa sem balanceamento

Prof. Douglas G. Macharet
douglas.macharet@dcc.ufmg.br

Introdução

- Árvore de pesquisa
 - Estrutura eficiente para armazenar informação
- Adequada quando da necessidade de
 - Acesso direto e sequencial eficientes
 - Facilidade de inserção retirada de registros
 - Boa taxa de utilização de memória
 - Utilização de memória primária e secundária

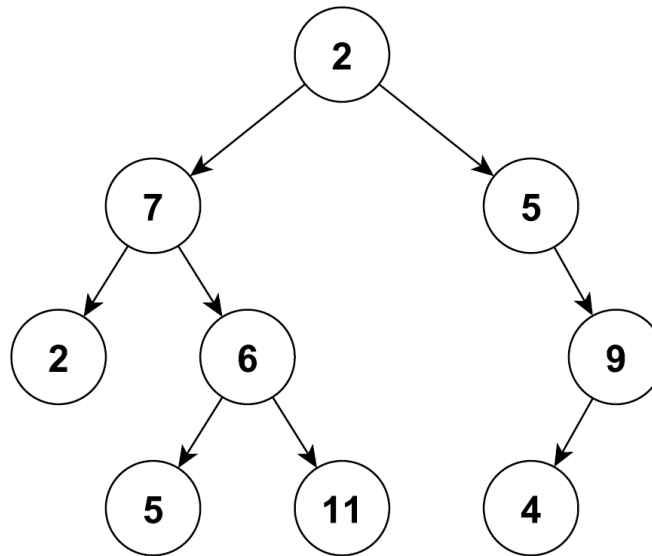
Árvore

- Estrutura hierárquica
 - Muito utilizada para organização de elementos
- Cada elemento é chamado de nó
- Relação: Pai – Filhos
 - Raiz: primeiro nó da hierarquia
 - Folhas: nós que não possuem filhos
- Filho de um nó é a raiz de uma subárvore

Árvore

Binária

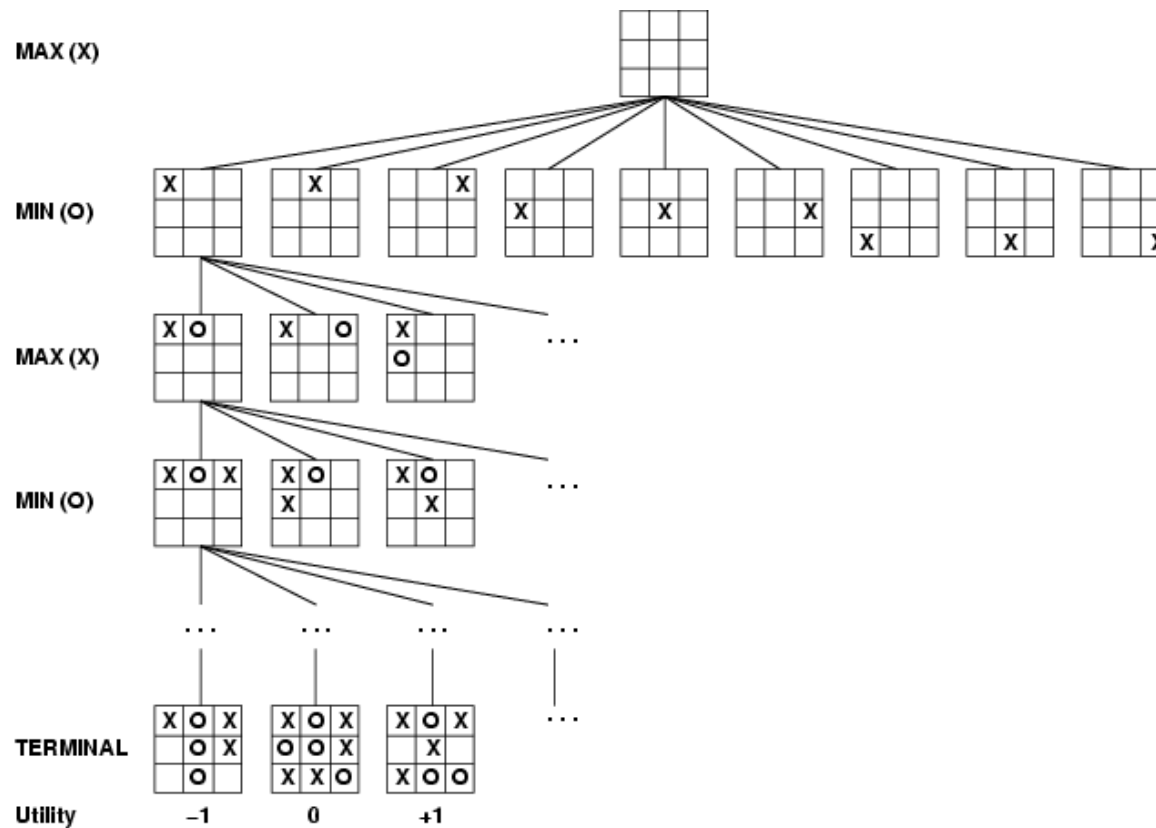
- Cada nó possui no máximo dois nós filhos



- Obs: a árvore acima não impõe nenhuma ordenação em seus nós!

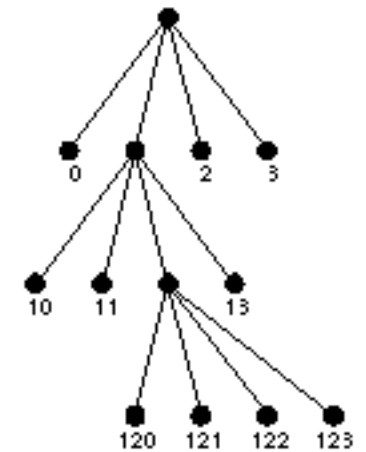
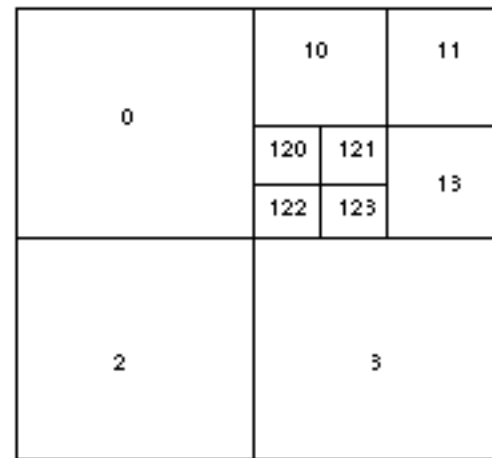
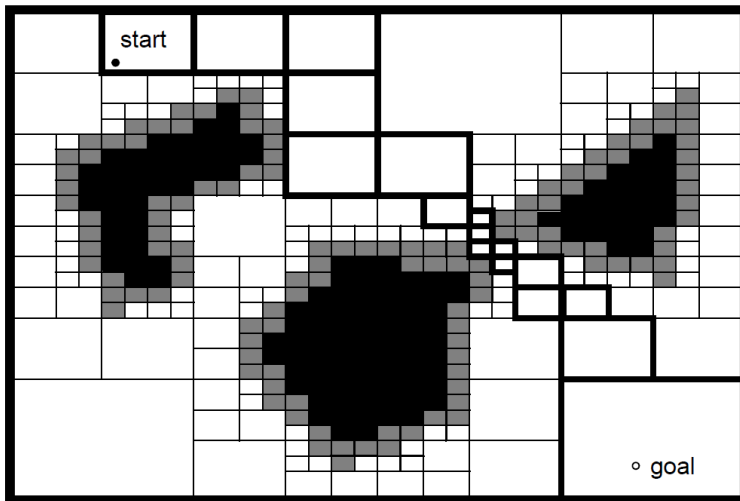
Árvore

Exemplo



Árvore

Exemplo



Árvore

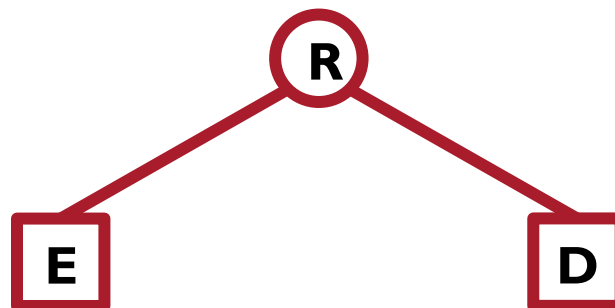
Exemplo



Árvore

Binária

- Até o momento consideramos árvores que apenas organizam os dados hierarquicamente
 - Não existe nenhuma restrição/inspeção
- Para qualquer nó que tenha um registro



Árvores Binárias de Pesquisa

- Tem-se a seguinte relação invariante

$$\boxed{E} < \textcircled{R} < \boxed{D}$$

- Todos os registros com chaves menores estão dispostos na subárvore à esquerda
- Todos os registros com chaves maiores estão dispostos na subárvore à direita

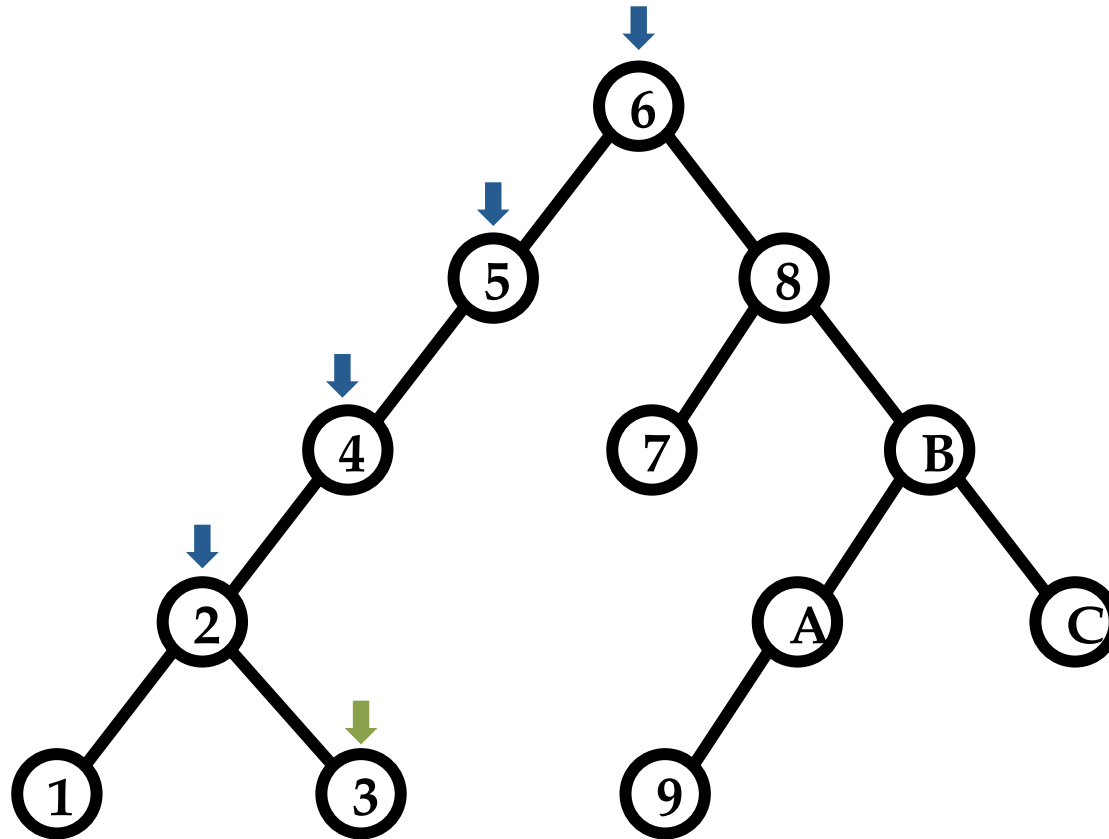
Árvores Binárias de Pesquisa

Procedimento para pesquisar

- Para encontrar um registro com chave x
 - Compare-a com a chave que está na raiz
 - Se x é menor, vá para a subárvore da esquerda
 - Se x é maior, vá para a subárvore da direita
 - Repita o processo recursivamente, até a chave ser encontrada ou um nó folha seja alcançado
 - Em caso de sucesso, retorna o próprio registro

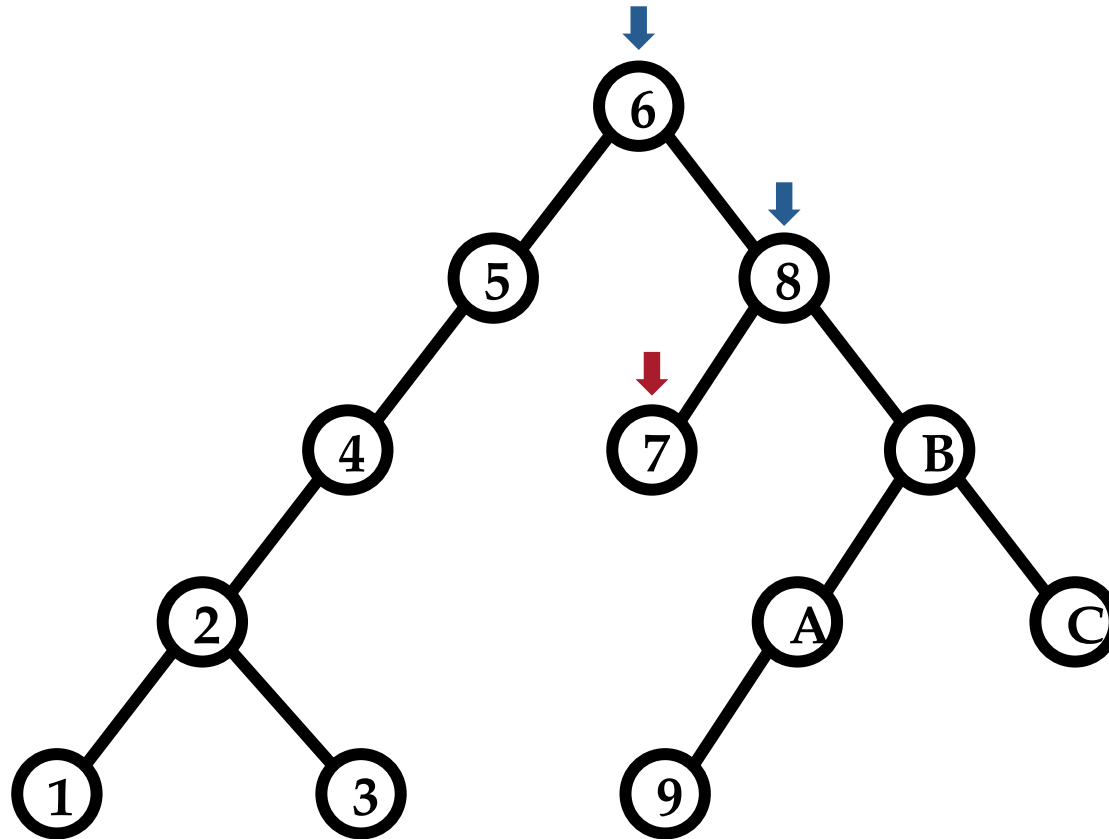
Árvores Binárias de Pesquisa

Pesquisa (Com sucesso – Busca pela chave 3)



Árvores Binárias de Pesquisa

Pesquisa (Sem sucesso – Busca pela chave 7,5)



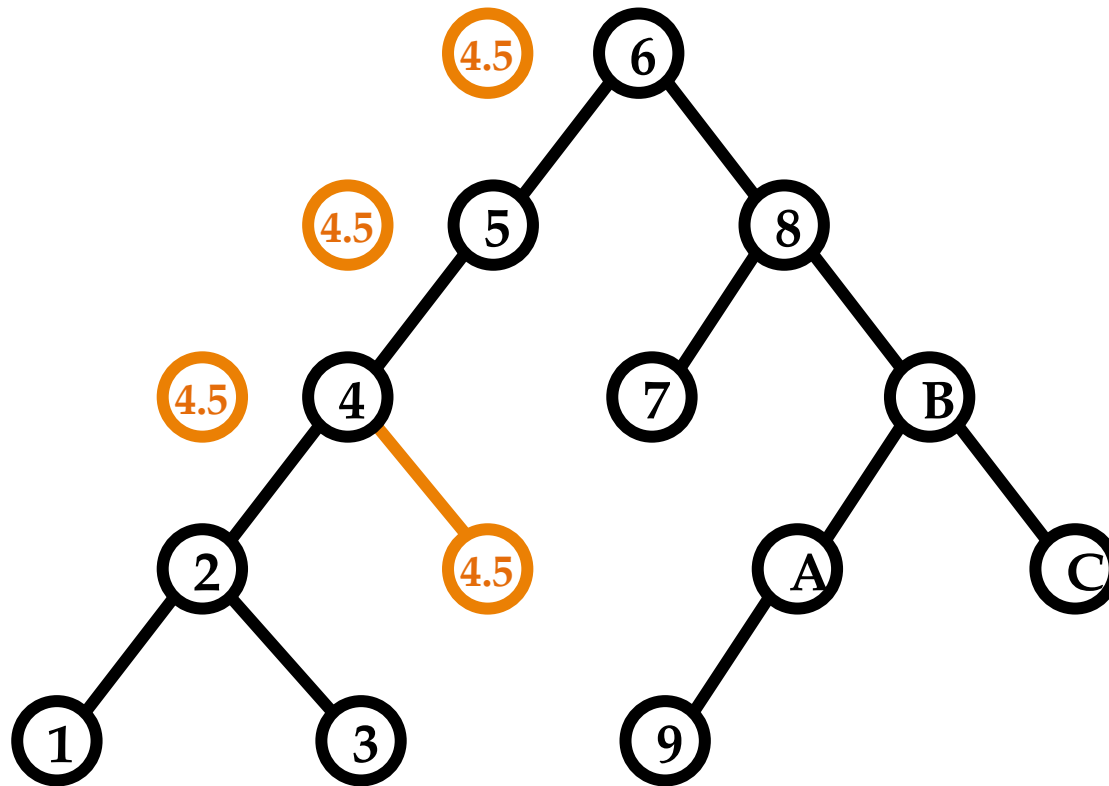
Árvores Binárias de Pesquisa

Procedimento para inserir

- Pesquisa sem sucesso
 - Atingir um apontador nulo durante a pesquisa
- O apontador nulo será o ponto de inserção
- Observação
 - O apontador deve ser passado por referência para a ligação correta do novo nó à árvore

Árvores Binárias de Pesquisa

Inserir



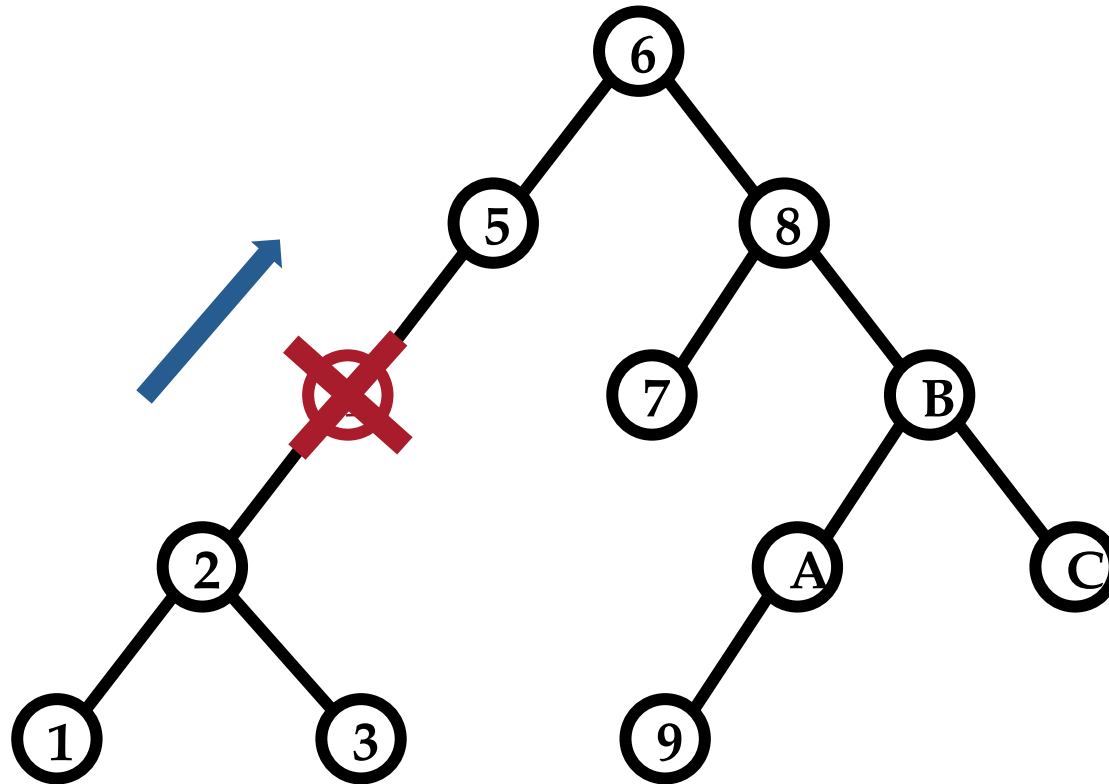
Árvores Binárias de Pesquisa

Procedimento para retirar

- Retirar não é tão simples quanto inserir
- Nó que contém o registro possui um filho
 - Operação simples
- Nó que contém o registro possui dois filhos
 - Substituí-lo com registro mais à direita na subárvore da esquerda; ou pelo registro mais à esquerda na subárvore da direita

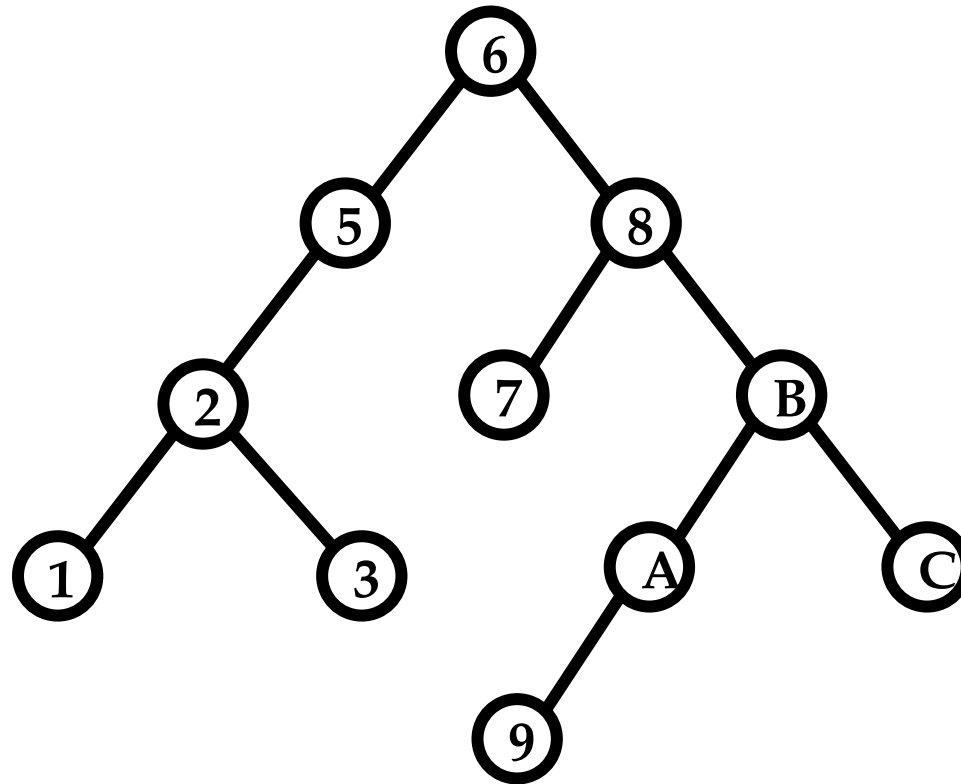
Árvores Binárias de Pesquisa

Retirar (Um filho)



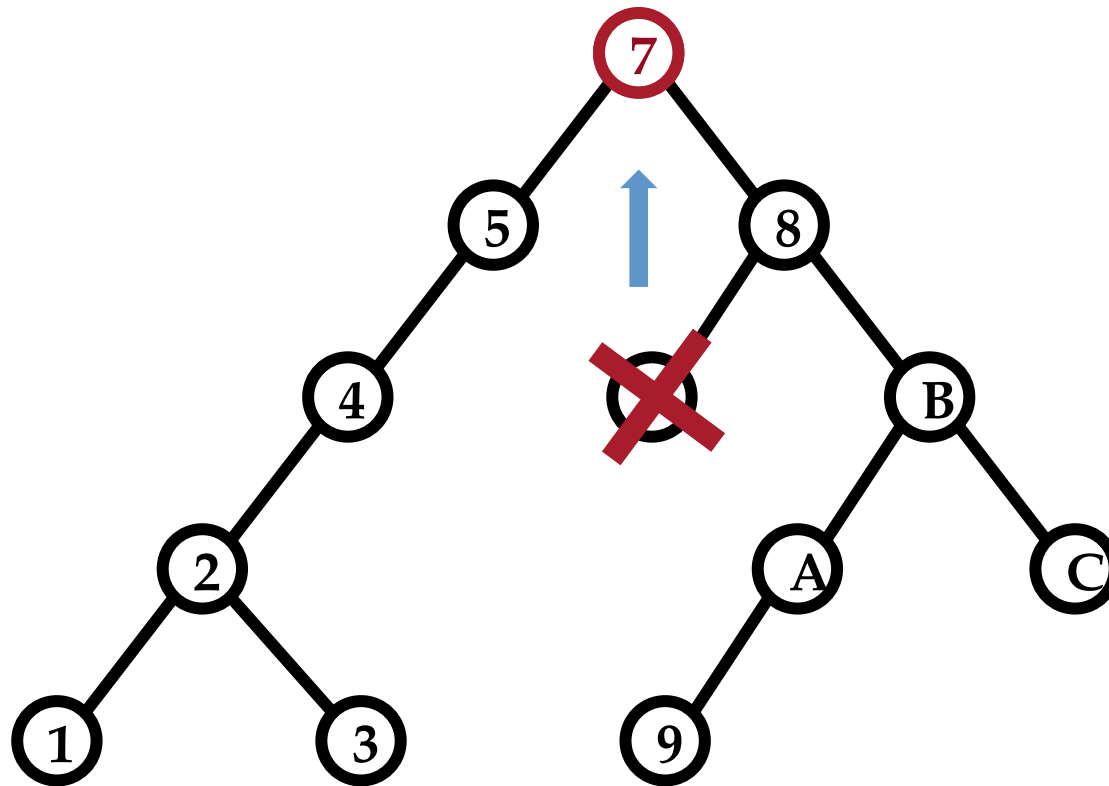
Árvores Binárias de Pesquisa

Retirar (Um filho)



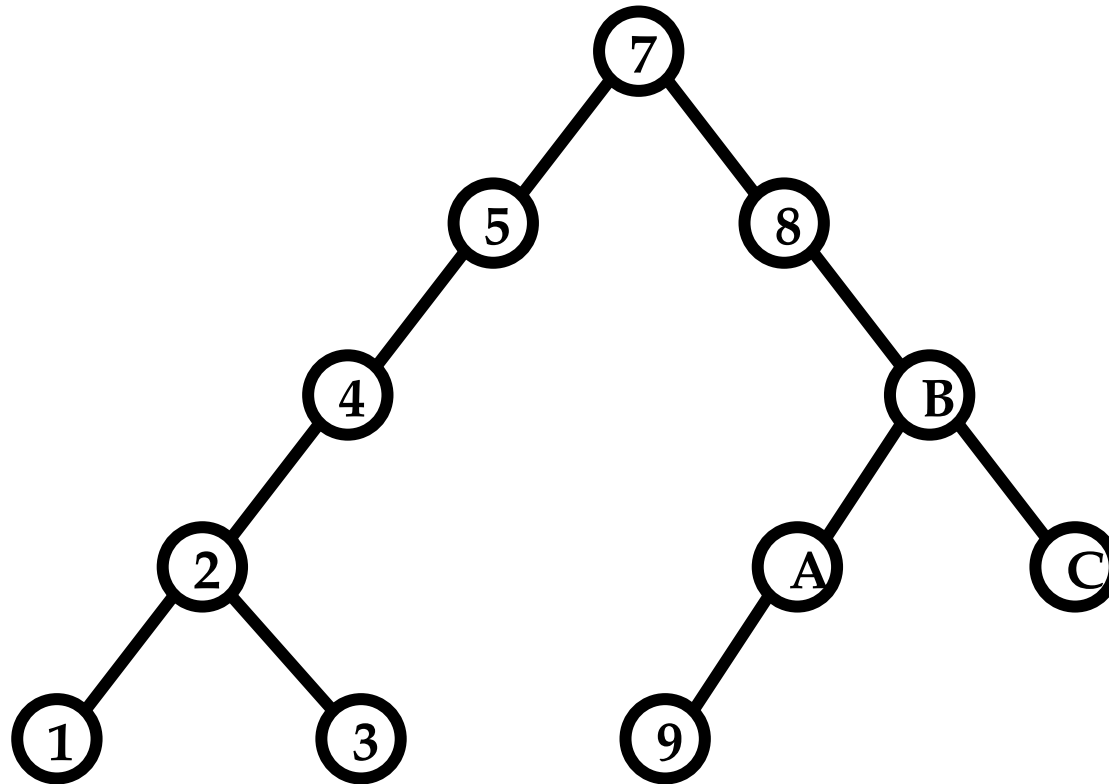
Árvores Binárias de Pesquisa

Retirar (Dois filhos) – Min/Sucessor



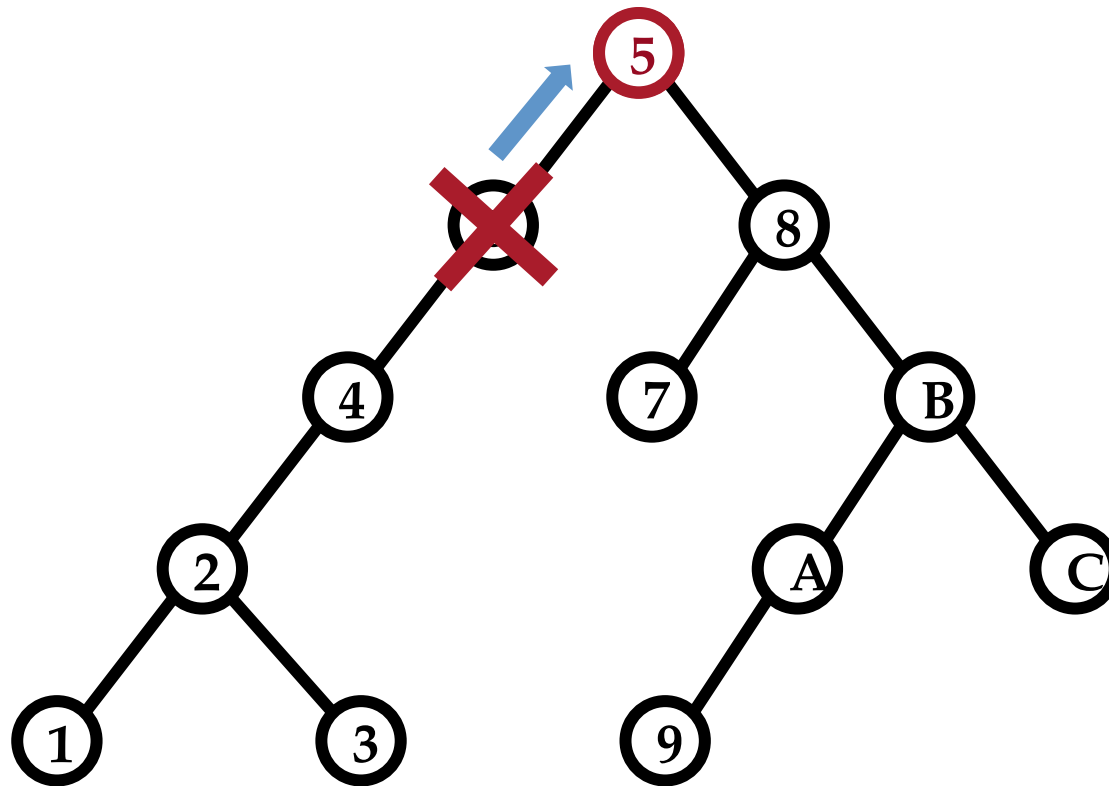
Árvores Binárias de Pesquisa

Retirar (Dois filhos) – Min/Sucessor



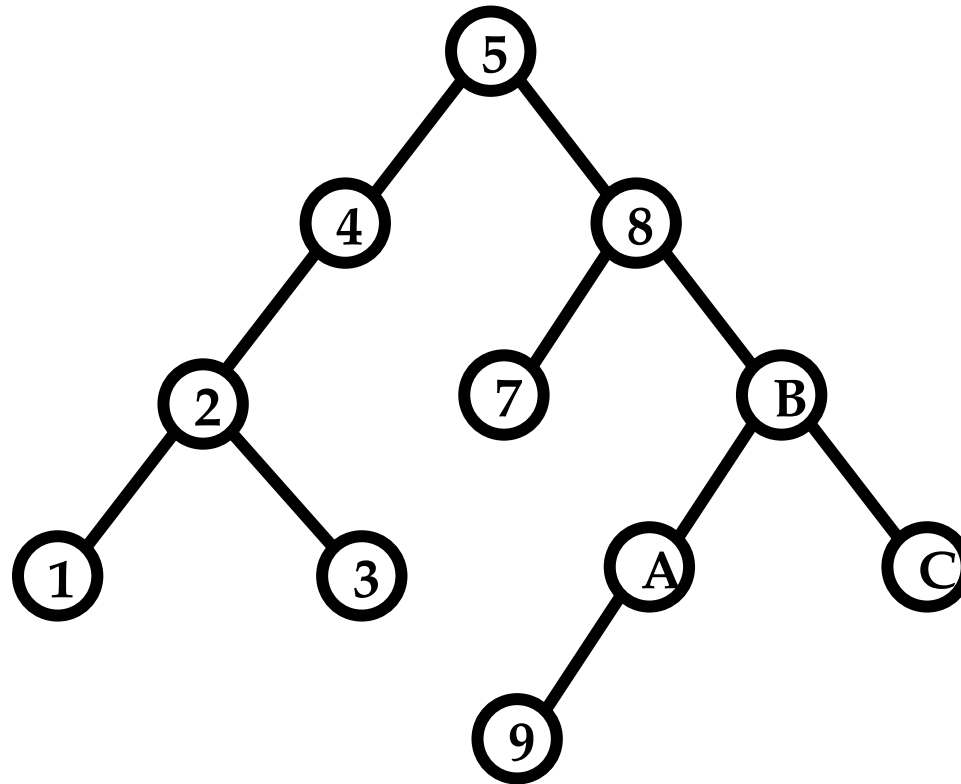
Árvores Binárias de Pesquisa

Retirar (Dois filhos) – Max/Antecessor



Árvores Binárias de Pesquisa

Retirar (Dois filhos) – Max/Antecessor



Árvores Binárias de Pesquisa

Caminhamento

- Como percorrer os nós das árvores?
 - Diferentes maneiras (sequências)
- Caminhamentos
 - Pré-Ordem
 - Central
 - Pós-Ordem

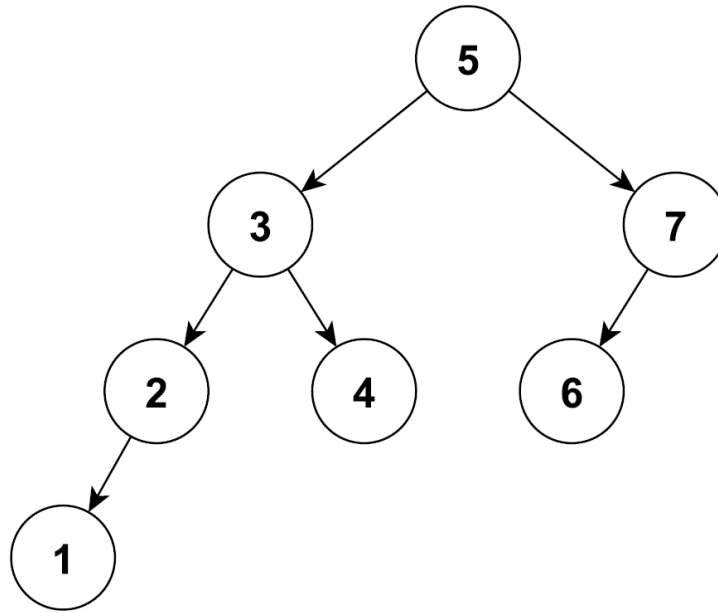
Árvores Binárias de Pesquisa

Caminhamento central

- Caminhamento
 - Caminha na subárvore esquerda
 - Visita a raiz
 - Caminha na subárvore direita
- Nós visitados de forma ordenada

Árvores Binárias de Pesquisa

Caminhamento central



- Ordem de visita: 1, 2, 3, 4, 5, 6, 7

Árvores Binárias de Pesquisa

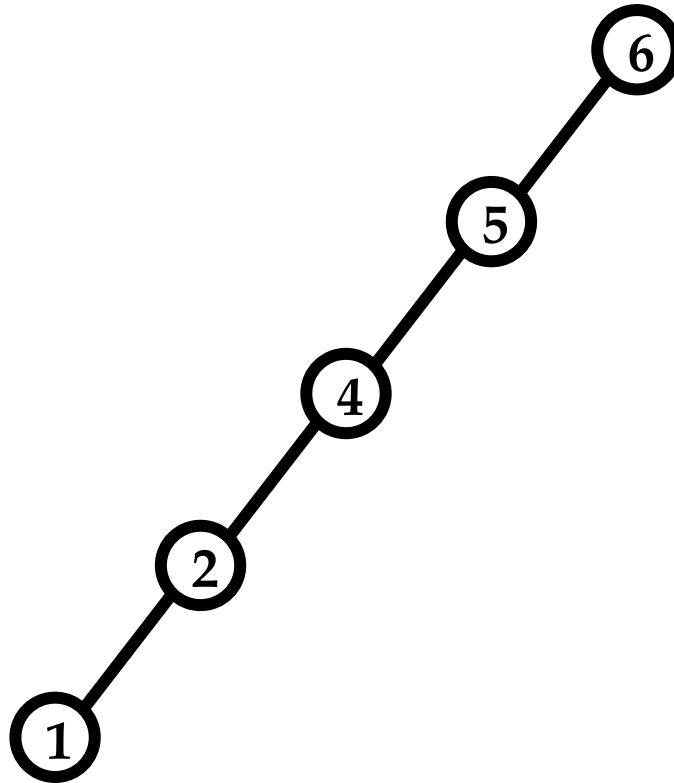
Análise

- Pesquisa com sucesso
 - Melhor caso: $\mathcal{C}(n) = O(1)$
 - Pior caso: $\mathcal{C}(n) = O(n)$
 - Caso médio: $\mathcal{C}(n) = O(\log n)$
- O tempo de execução para árvores binárias de pesquisa dependem muito do formato

Árvores Binárias de Pesquisa

Análise

- Ordem de remoção/inserção influencia



Árvores Binárias de Pesquisa

Análise

- Pior caso
 - Chaves em ordem crescente/decrescente
 - Lista linear
 - Custo médio de comparações: $n+1/2$
- Árvore de pesquisa aleatória
 - Probabilidade igual de inserção em qualquer nó
 - Comparações esperadas: $1,39\log n$

Árvores Binárias de Pesquisa

Análise

- É possível resolver esse problema?
 - Como?
- Árvores Totalmente (Parcial) Balanceadas