



BST

BINARY SEARCH TREE

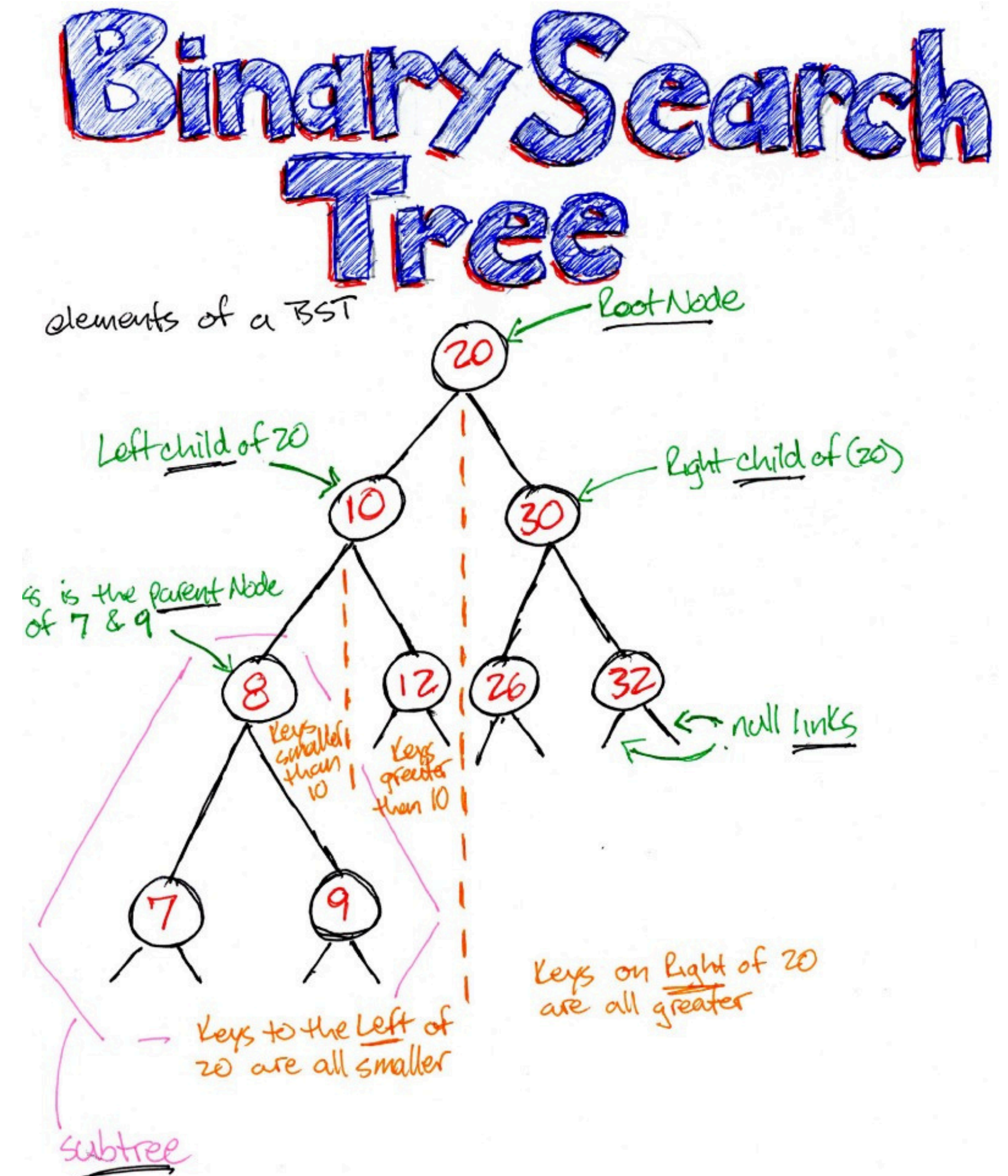
Bruno César Silva de Souza
Larissa Silva de Souza Neitzke

O QUE É UMA BINARY SEARCH TREE (BST)?

- Estrutura de dados em forma de árvore binária;
- Cada nó tem no máximo dois filhos: **esquerda** (valores menores) e **direita** (valores maiores);
- Baseada no princípio da busca binária.

COMO FUNCIONA?

- Busca começa pela raiz;
- Compara valor:
 - Se menor → vai para a **esquerda**.
 - Se maior → vai para a **direita**.
- Continua até encontrar o valor ou chegar a um nó nulo.



OPERAÇÕES BÁSICAS

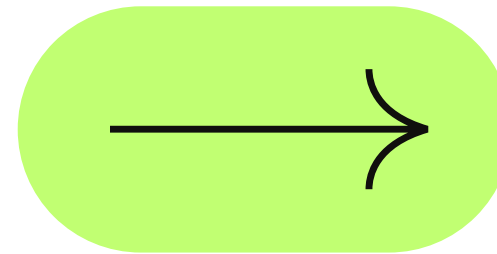
- **Inserir:** encontra posição certa mantendo a ordem;
- **Pesquisar:** percorre os nós comparando os valores;
- **Excluir:** trata 3 casos: sem filhos, com 1 filho ou 2 filhos;
- **Travessias:**
 - In-ordem: **Esquerda → Nó → Direita**
Útil para listar os elementos ordenados.
 - Pré-ordem: **Nó → Esquerda → Direita**
Útil para copiar ou salvar a árvore.
 - Pós-ordem: **Esquerda → Direita → Nó**
Útil para deletar todos os nós da árvore.

Comparação entre Tipos de Árvores Binárias

Tipo de Árvore	Estrutura	Tempo Médio de Busca	Pior Caso de Busca	Balanceamento	Ideal para...
BST (básica)	Binária, filhos à esquerda < pai < direita	$O(\log n)$	$O(n)$	✗ Não garantido	Casos simples, estrutura leve, poucos dados
AVL	Binária, com balanceamento rígido	$O(\log n)$	$O(\log n)$	✓ Muito bem balanceada	Consultas rápidas e frequentes
Rubro-Negra	Binária, balanceamento flexível com cores	$O(\log n)$	$O(\log n)$	✓ Balanceada	Árvores grandes, sistemas como bancos de dados
B-Tree	Multi-filhos (mais que 2 por nó)	$O(\log n)$	$O(\log n)$	✓ Muito balanceada	Armazenamento em disco, bancos de dados

LIMITAÇÕES

Pode ficar desbalanceada,
virando uma lista.



O tempo de busca
pode piorar para $O(n)$.

Solução: usar árvores
autobalanceadas
(AVL, Rubro-Negra).



Agradecemos a atenção!