

Árvores binárias de busca

Terminologia

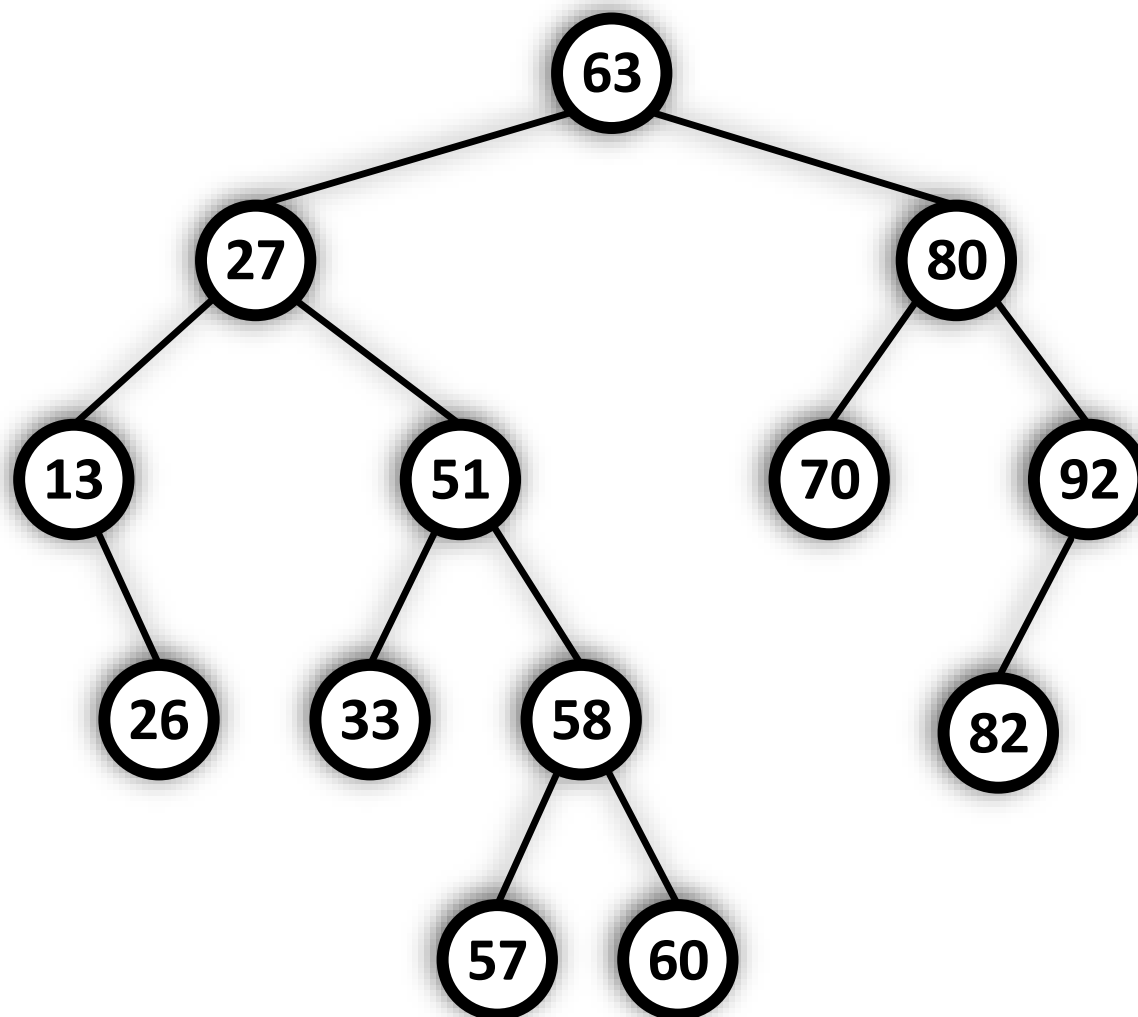
- **Chave de busca**

- É um valor utilizado para buscar um item numa árvore
- Em Java, a chave de busca é implementada pelo método `equals()`.

Árvores binárias de busca

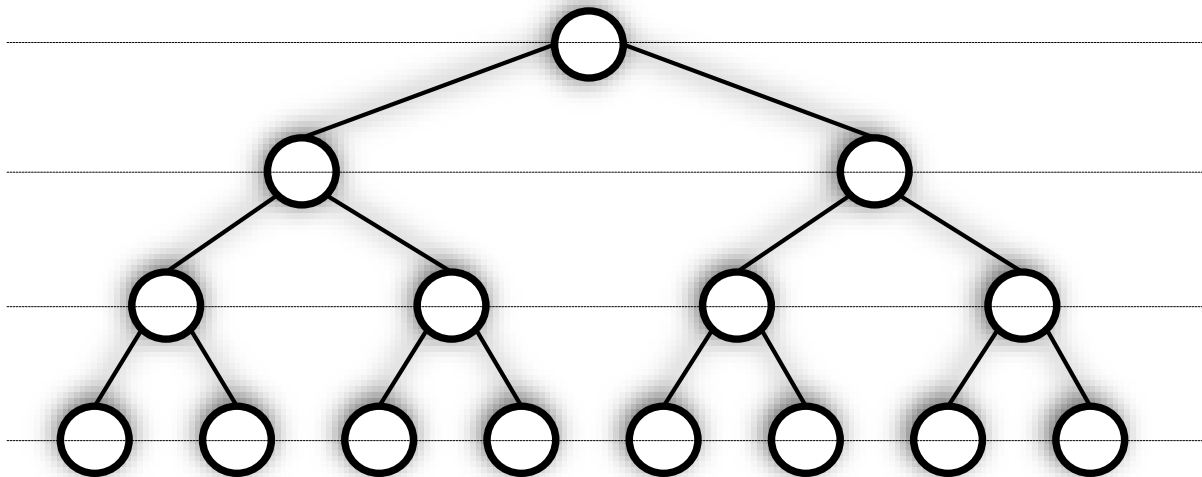
- É uma árvore binária que possui as seguintes características:
 - O filho à esquerda de um nó possui chave menor que seu pai
 - O filho à direita de um nó possui uma chave maior ou igual ao seu pai

Exemplo de árvore de busca



Árvore cheia

- Numa **árvore cheia** todos os nós internos tem duas sub-árvores associadas
- Numa árvore cheia, o número de nós numa árvore binária de altura **h** é igual à **$2^{h+1} - 1$**



Nível 0: $2^0 = 1$ nó

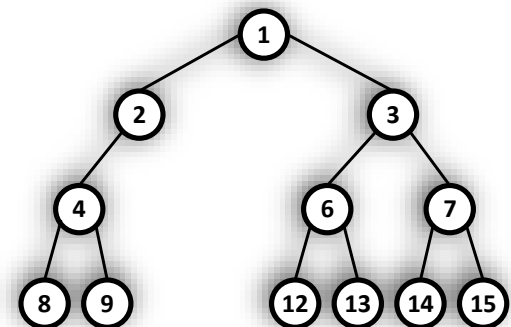
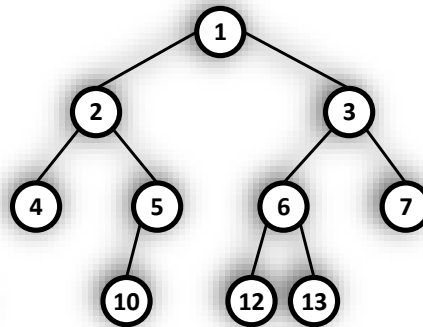
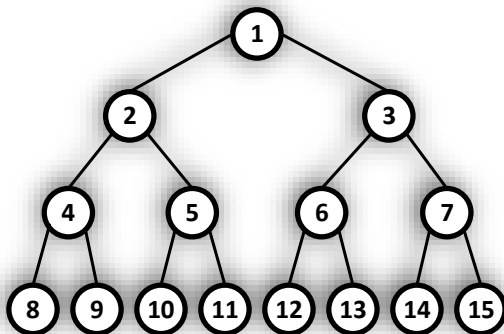
Nível 1: $2^1 = 2$ nós

Nível 2: $2^2 = 4$ nós

Nível 3: $2^3 = 8$ nós

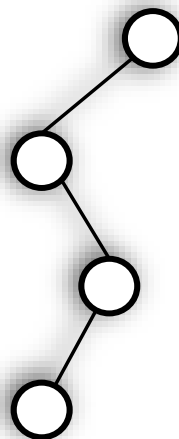
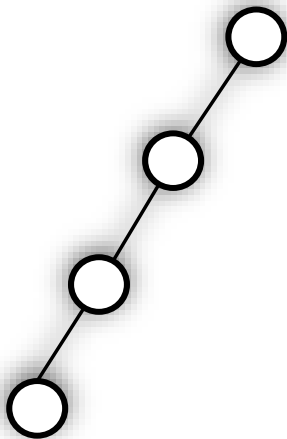
Árvore Balanceada

- Uma árvore binária está balanceada quando as alturas das duas sub-árvores de cada um dos nós nunca diferem em mais de 1 unidade
- Exemplos de árvores balanceadas:
- Exemplo de árvore não balanceada:



Árvore degenerada

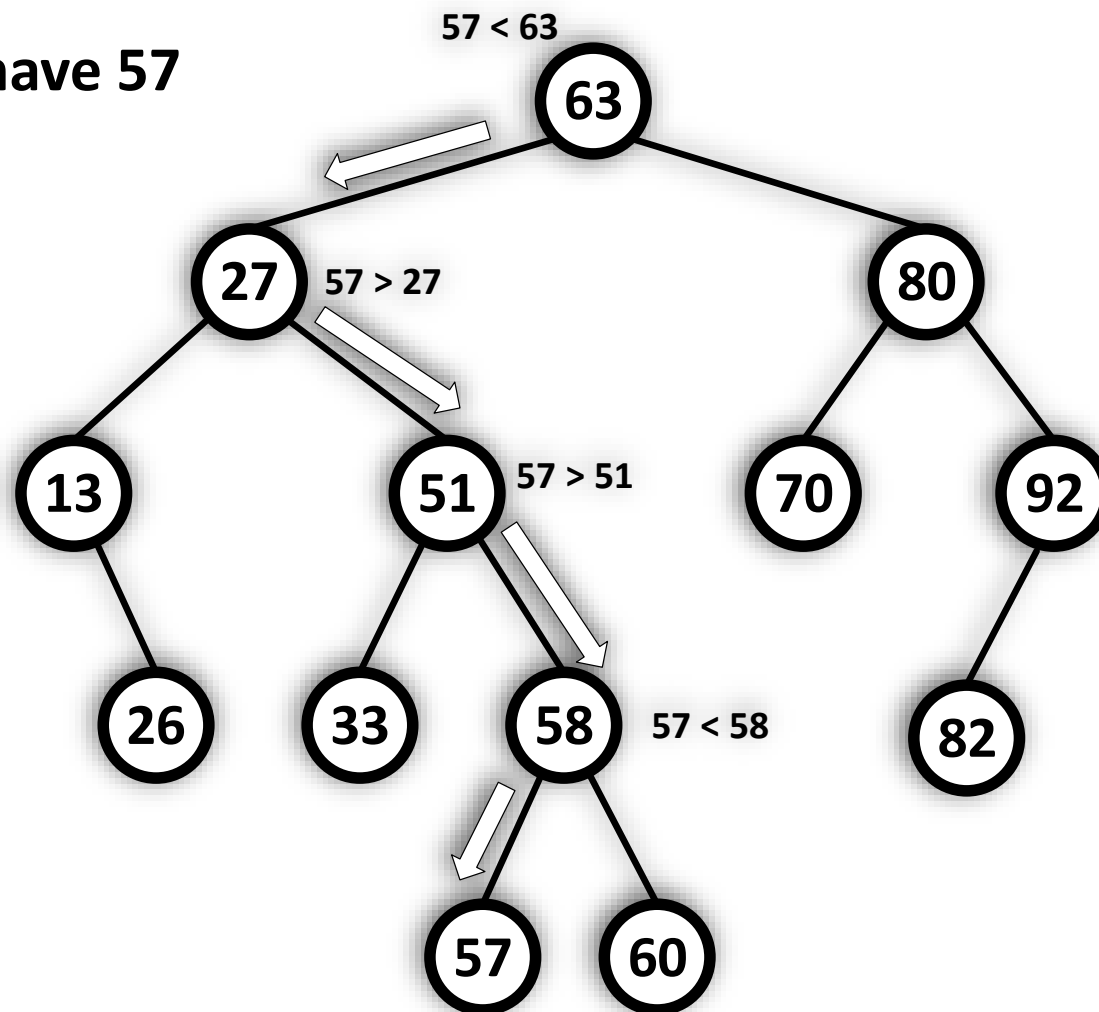
- Uma árvore é degenerada quando todos os nós internos têm uma única sub-árvore não vazia associada
- O número de nós (n) de uma árvore binária degenerada de altura h é $h + 1$



$h=3 \rightarrow n=4$

Busca na árvore binária

Buscar chave 57



Localização de um nó na árvore

(algoritmo recursivo)

Algoritmo: **buscar**(NoArvore no, int valor)

se (no = null) **então**

retornar null;

fim-se

se (valor = no.info) **então**

retornar no;

senão

se (valor < no.info) **então**

retornar buscar(no.esquerda, valor);

senão

retornar buscar(no.direita, valor);

fim-se

fim-se

Localização de um nó na árvore

(algoritmo iterativo)

Algoritmo: **buscar**(int valor)

p ← raiz;

enquanto (p ≠ null) e (p.info ≠ valor) **faça**

se (valor < p.info) **então**

 p ← p.esquerda;

senão

 p ← p.direita;

fim-se;

fim-se

retornar p;

Algoritmo: **inserir(int valor)**

NoArvoreBinaria novo = new NoArvoreBinaria(valor);

Se (raiz = null) **então**

raiz \leftarrow novo;

senão

p \leftarrow raiz;

enquanto (verdadeiro) **faça**

pai \leftarrow p;

se (valor < p.info) **então**

p \leftarrow pai.esquerda;

se (p = null) **então**

pai.esquerda \leftarrow novo;

retornar;

fim-se;

senão

p \leftarrow pai.direita;

se (p = null) **então**

pai.direita \leftarrow novo;

retornar;

fim-se;

fim-se;

fim-enquanto;

fim-se;

Inserção de nó

Algoritmo iterativo
para inserir um nó
numa árvore binária
de busca:

Localiza em que
posição deve ser
adicionado o dado,
similar ao algoritmo
de busca