

Estrutura de Dados II

Ciência da Computação

Prof. André Kishimoto
2023

Árvores

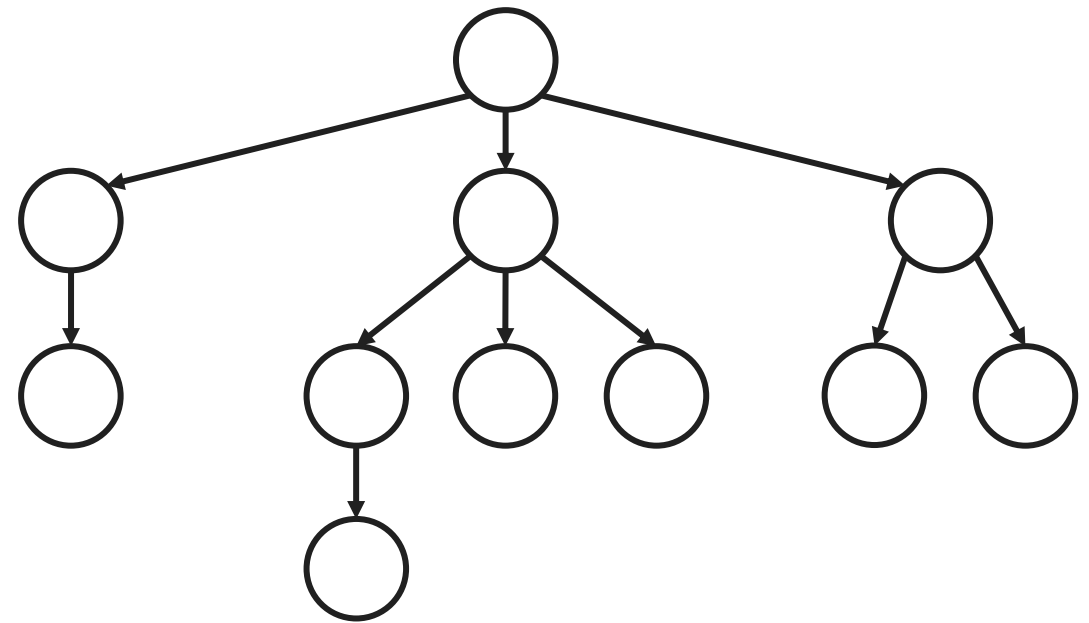
- Estrutura hierárquica.
 - Estrutura de dados não linear importante!
 - Eficiente para pesquisas.
- Adequada para representação de hierarquias.
 - Melhorar a organização de dados.
 - Facilitar o acesso aos dados.
 - Exemplo: *Como você armazena os trabalhos da faculdade ou as fotos que você tira com o celular na memória (disco) do computador? Existe alguma estrutura ou padrão?*

Árvores



/Users/akishimoto/Tree.bin

ex.: Brasil



Árvores

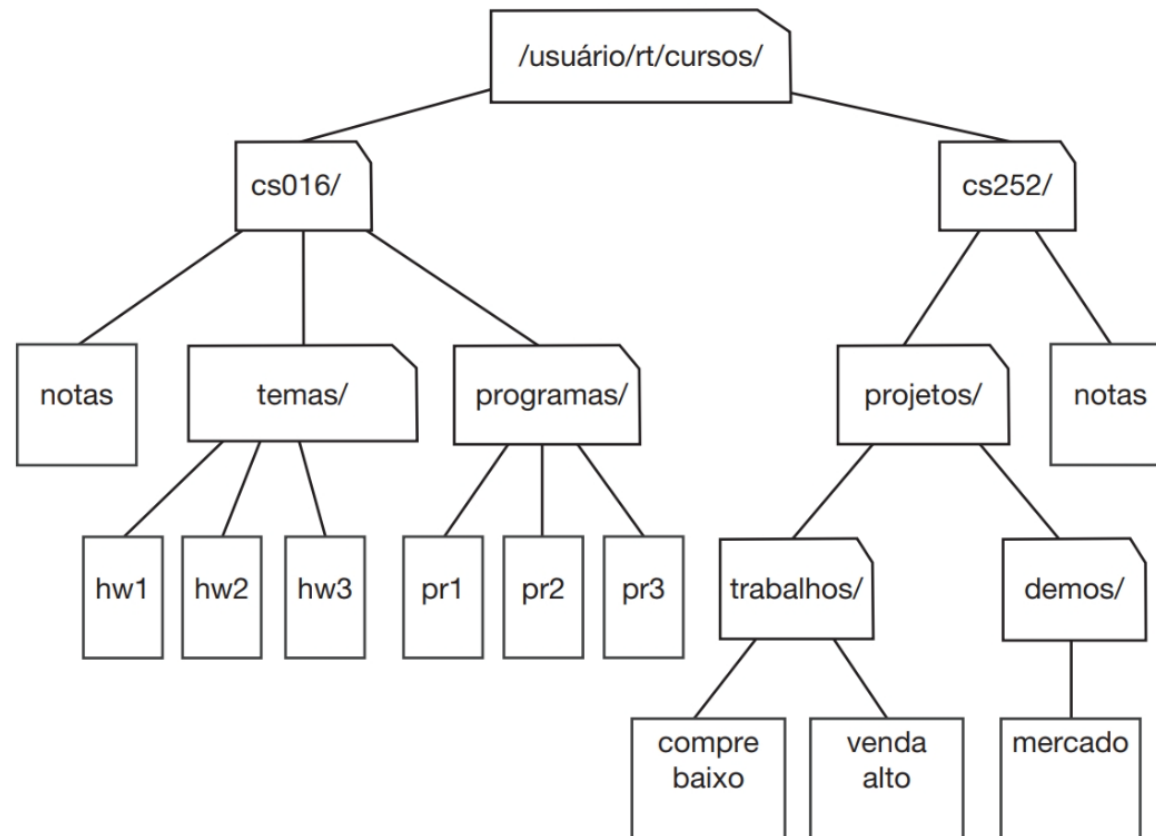


Figura 7.3 Árvore representando parte de um sistema de arquivos.

GOODRICH, M. T.; TAMASSIA, R. Estrutura de Dados e Algoritmos em Java, 5ª ed. Porto Alegre: Bookman, 2013.

Árvores

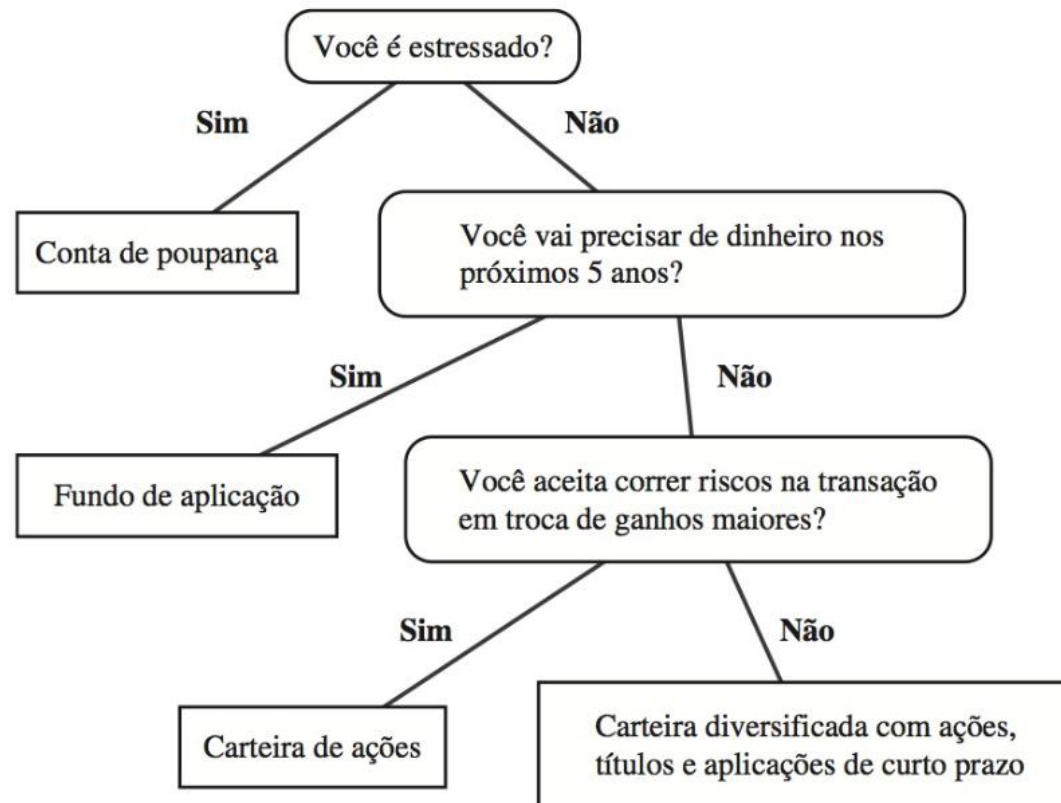


Figura 7.10 Árvore de decisão que fornece dicas de investimento.

GOODRICH, M. T.; TAMASSIA, R. Estrutura de Dados e Algoritmos em Java, 5ª ed. Porto Alegre: Bookman, 2013.

Árvores

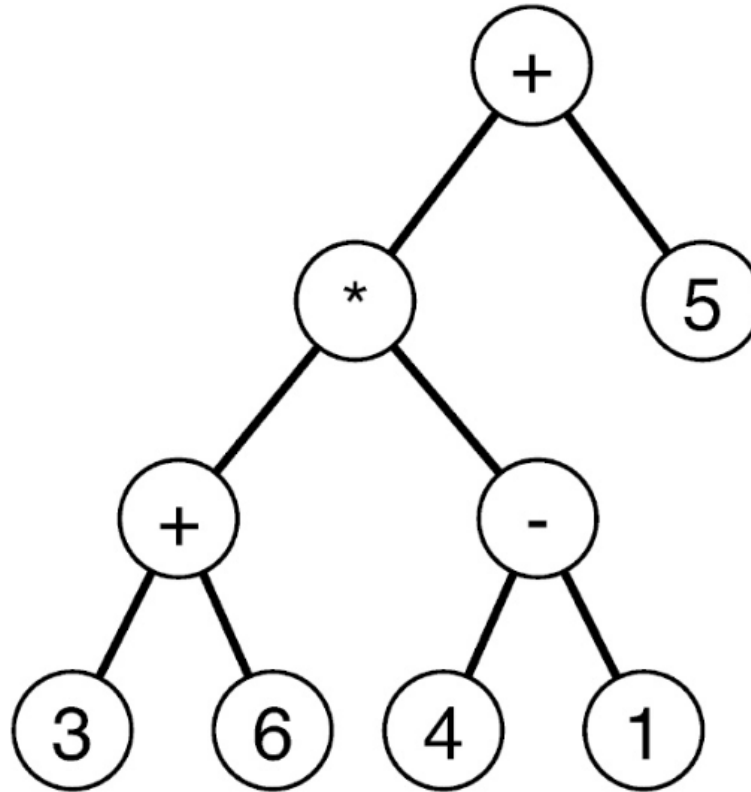


Figura 16.2: Árvore da expressão: $(3 + 6) * (4 - 1) + 5$.

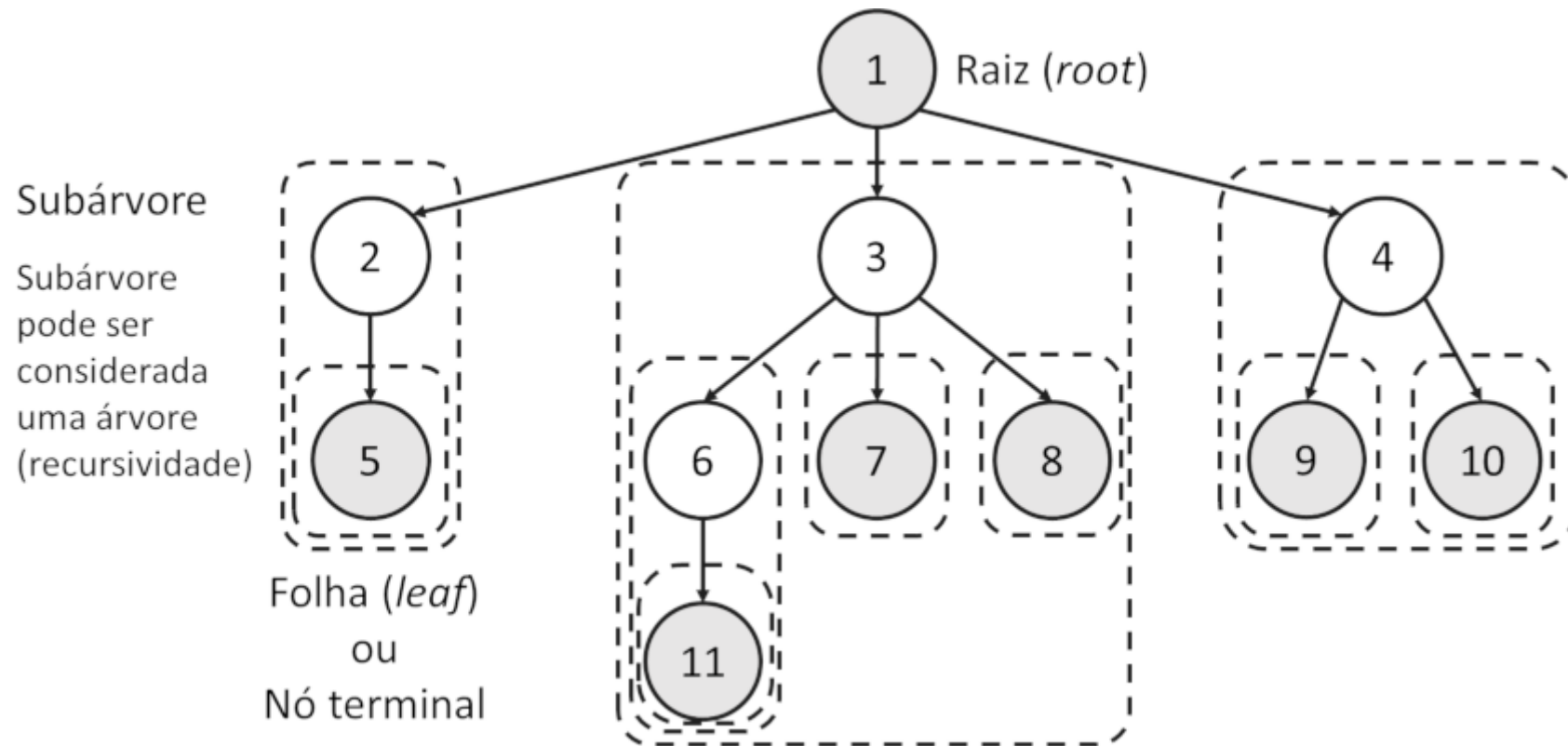
Árvores – conceitos

- Assim como listas, árvores são compostas por um conjunto de **nós** (os dados são armazenados em nós).
- Toda árvore (não-vazia) possui um nó denominado **raiz** (*root*).
- Todo nó de uma árvore é uma **subárvore** (*subtree*).
 - Recursividade!
- Relação de parentesco (hierarquia) entre os nós:
 - Nós raízes das subárvores são **filhos** (*children, child*) do nó **pai** (*parent*).
 - Nós só podem ter um único nó pai, mas podem ter 0+ filhos.
 - O único nó que não tem pai é o nó raiz.

Árvores – conceitos

- Nós que possuem filhos são chamados de **nós internos** (*internal nodes*).
- Nós que não possuem filhos são chamados de **folhas** (*leaves, leaf, leaf node*), **nós externos** (*external nodes*) ou **nós terminais** (*terminal nodes*).
- Nós com mesmo pai são **irmãos** (*siblings*).

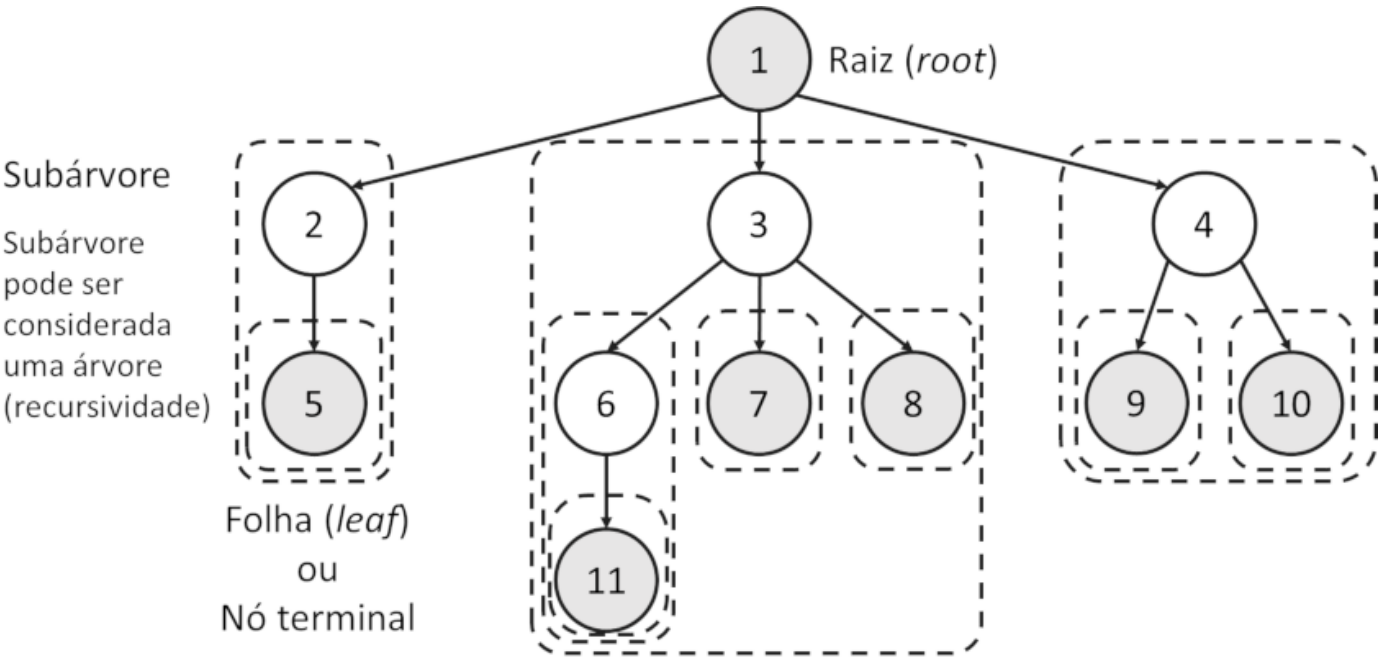
Árvores – conceitos



Árvores – grau

- Todo **nó** possui um **grau** (*degree*), definido pelo número total de filhos do nó.
- Toda **árvore** também possui um **grau**, definido pelo maior grau de um nó pertencente à árvore.

Árvores – grau



E o grau da árvore?

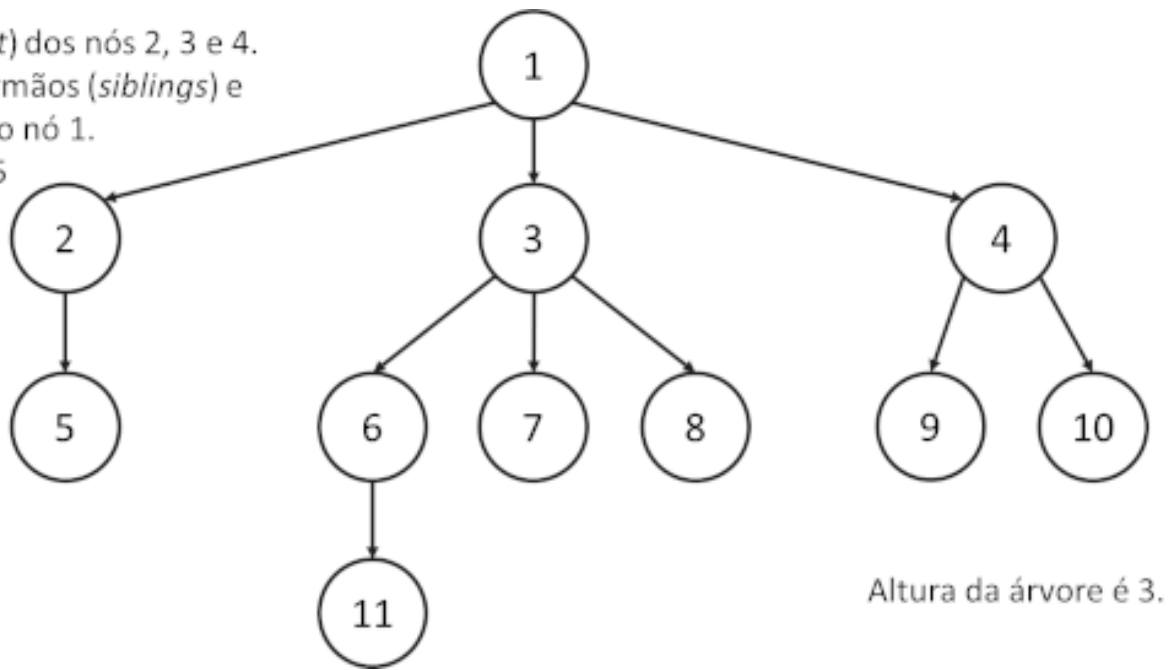
Nó	Grau do nó (quantidade de filhos)
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

Árvores – altura

- Toda árvore possui uma **altura** (*tree height*), definida pelo caminho mais longo do nó raiz até uma das folhas.
 - Podemos contar as **arestas** (*edges*) que existem no caminho mais longo da raiz até uma das folhas ou
 - Usar o maior **nível** da árvore (nível mais profundo da árvore) – ver adiante.
- Nós também possuem altura (um nó é uma subárvore).
 - **Folha** tem **altura zero** (há zero arestas abaixo desse nó).
 - Se não for folha, calculamos a altura de um nó com:
 $h(N) = \text{altura do nó } N = 1 + \text{maior altura dos nós filhos de } N$

Árvores – altura

Nó 1 é pai (*parent*) dos nós 2, 3 e 4.
Nós 2, 3 e 4 são irmãos (*siblings*) e filhos (*children*) do nó 1.
Nó 2 é pai do nó 5
(que é filho do nó 2).
Etc...



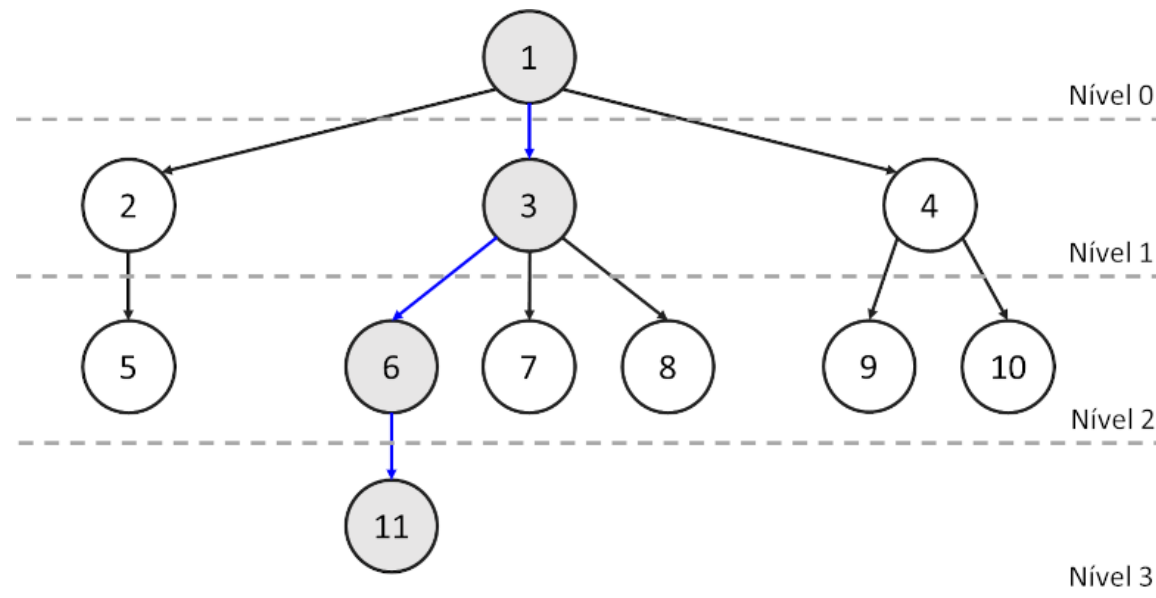
Altura da árvore é 3.

Nó	Altura do nó – h(n)
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

Árvores – nível dos nós

- Os nós de uma árvore possuem um **nível** (*level*, *depth*), que pode ser obtido com a equação:

$$d(N) = \text{nível do nó } N = \text{nível do pai de } N + 1$$



Exercício (para casa e “participação”)

Faça uma pesquisa sobre a estrutura de dados Árvore e descreva, de forma resumida, pelo menos três aplicações dessa estrutura de dados e um exemplo real de software ou biblioteca que usa o TAD árvore (para cada aplicação). Inclua o tipo de árvore usada em cada exemplo.

Não esqueça de incluir as referências usadas na pesquisa!

Exemplo:

Aplicação 1 – XYZ

No cenário XYZ, o TAD árvore é usado para isso e aquilo. Um exemplo dessa aplicação pode ser vista no software ABC.

Referências:

(1) ...

(2) ...