

Estrutura de dados II

Prof. Fabio Gomes Rocha

prof.fabiorocha@ymail.com

www.fgrweb.com.br

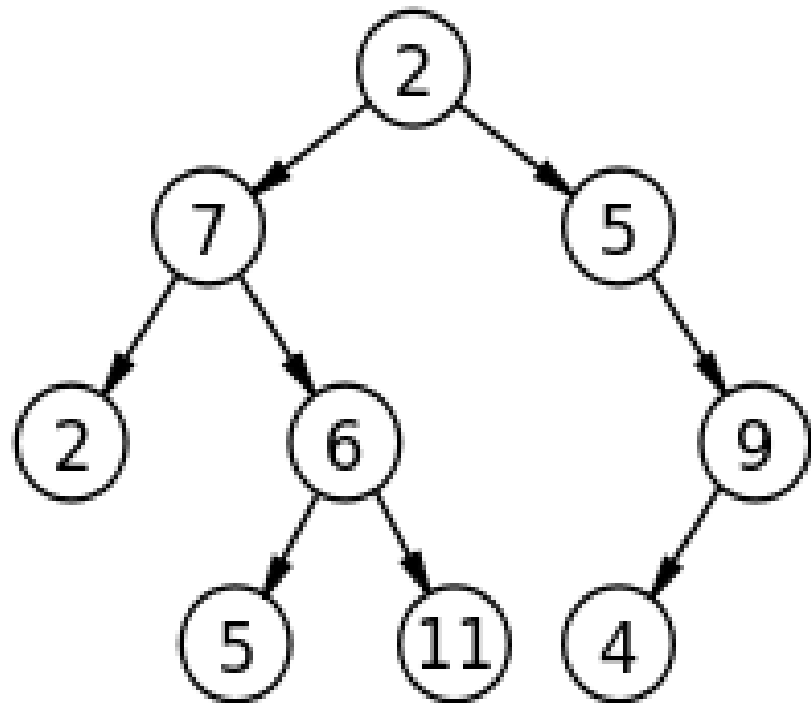
Arvore

Revisão

Vimos na aula passada que uma árvore é uma estrutura não linear que representa hierarquia, sendo que esta possui um conjunto de nós.

Vimos ainda que o nó inicial, ou nó R é denominado raiz e contém zero ou mais subárvores, cujas raízes são ligadas diretamente a R.

Uma raiz se liga a um ou mais elementos, sendo que cada uma destas, forma uma nova subárvore, esses elementos são seus filhos.



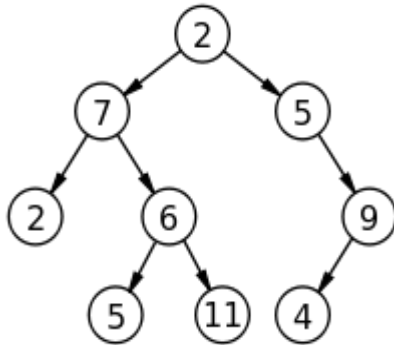
Árvore

Grau --> Número de subárvores de um nó

Folha --> nó que possui grau zero, ou seja, não possui arvores.

Filhos → são raízes das subarvores de um nó.

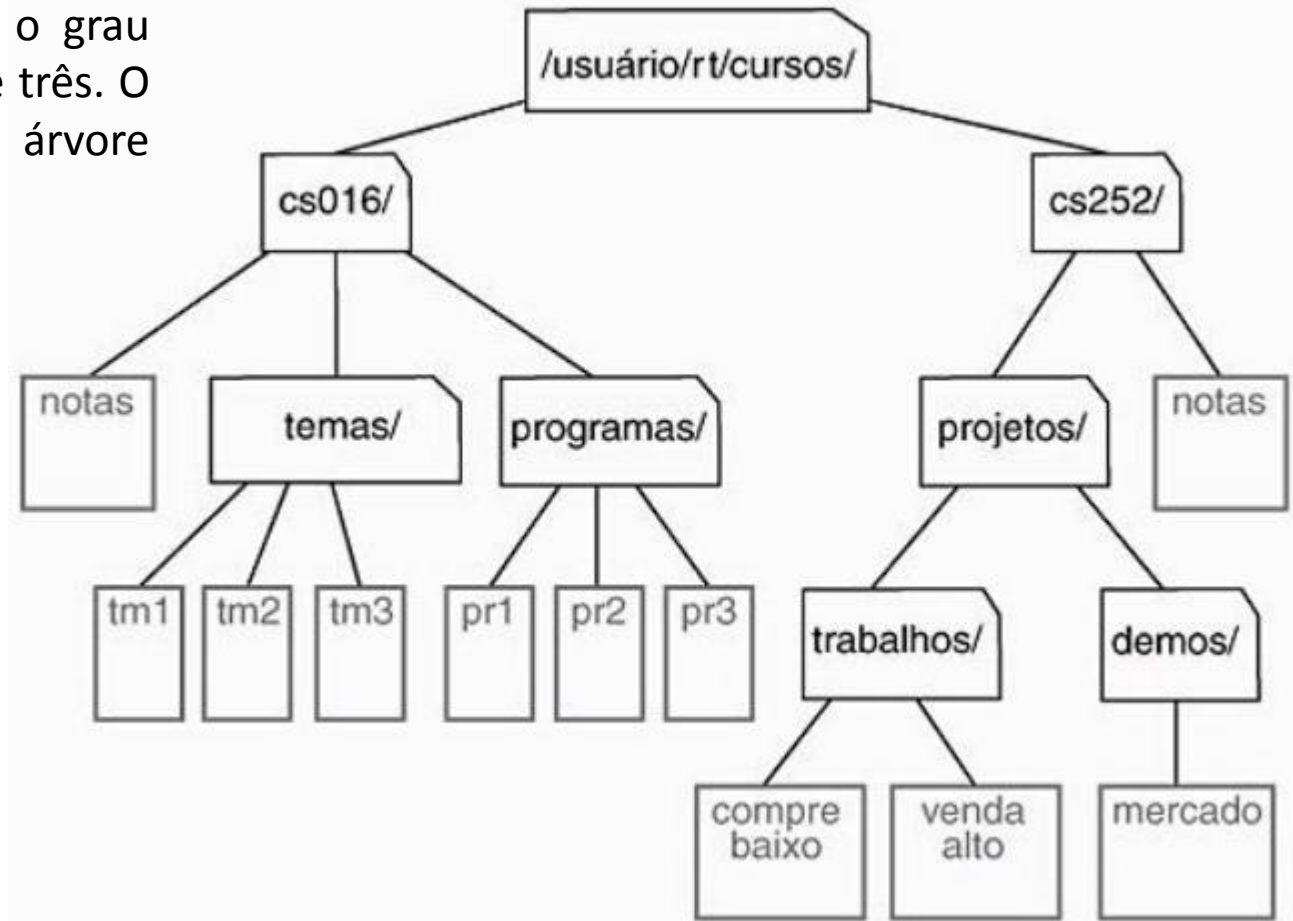
Altura → profundidade de uma árvore: É o máximo nível de seus nós



- Existem algumas divergências para adotar o grau de uma árvore, tal como existem divergências para um consenso sobre os níveis de uma árvore. Pesquisando iremos encontrar a mais usual e comum. Como segue:
 - O grau de um nó é definido pelo número de sub-árvores que ele possui.
 - O grau da árvore é definido pelo nó, de maior grau, que ela possui.
 - Exemplo de árvore binária:
 - Considerando a árvore binária ao lado, podemos considerar que ela é uma árvore de grau 2, uma vez que o maior grau de um ou mais nós que ela possui é 2.

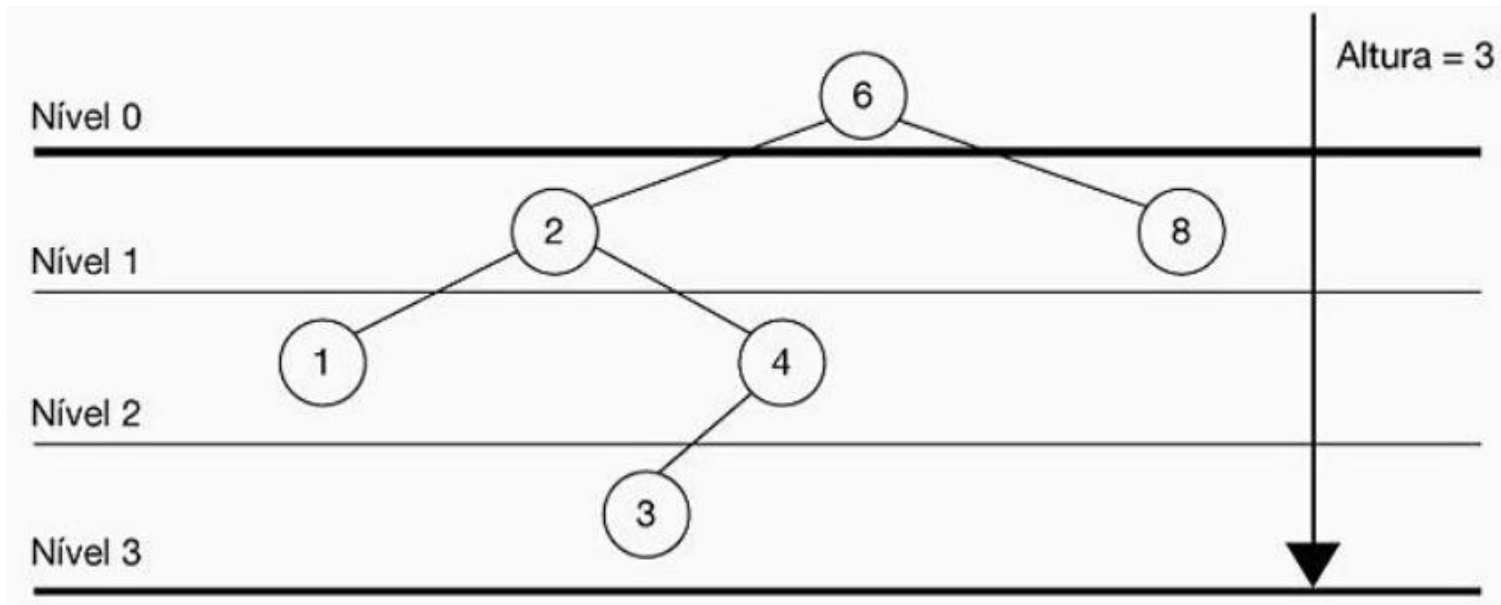
Árvore

No exemplo ao lado, o grau da árvore programas é três. O Grau máximo da árvore também é três.



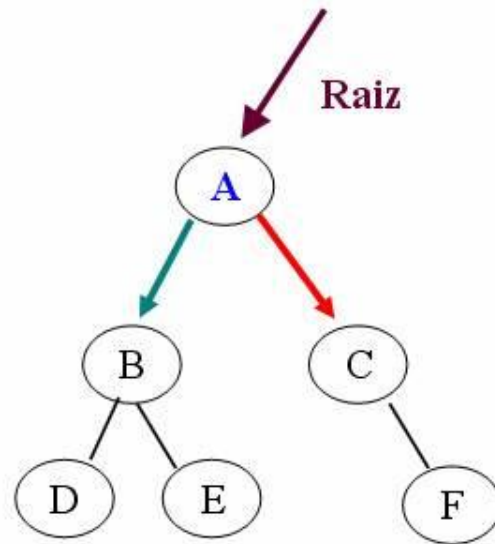
Árvore

A altura de uma árvore é o comprimento do caminho mais longo da raiz até uma das folhas, assim, uma árvore é considerada nula quando tem altura 0



Arvore

Todos os nós são acessíveis a partir da raiz, sendo que existe um único caminho entre a raiz e qualquer outro nó.



Terminologia:

- Filho esquerdo
- Filho direito
- Informação

Arvores

- As funções padrões da estrutura arvore são:
 - Elemento() → Retorna o objeto nesta posição
 - Root() → Retorna a raiz; um erro ocorre se a árvore esta vazia;
 - Parente(v) → retorna o nodo pai de v; ocorre erro se v for a raiz;
 - Children(v) → retorna uma coleção iterável contendo os filhos do nodo v;
 - Lembrando que se v é um nodo externo (folha) então children(v) esta vazio.
 - isInternal(v) → verifica se v é interno
 - isExternal(v) → verifica se v é externo
 - isRoot(v) → verifica se v é a raiz
 - size() → numero de nodos na árvore
 - isEmpty() → testa se a arvore tem algum nodo;
 - iterator() → retorna um iterador de todos os elementos nos nodos da árvore;
 - Positions() → Retorna uma coleção iterável com com todos os nodos da árvore;
 - Replace(v,e) → troca o elemento de v por e;

Arvores

- No iterator:
 - A interface Iterator define um tipo para acessar os elementos de uma determinada coleção, normalmente possuindo os seguintes métodos genéricos
 - hasNext()
 - determina se há um próximo elemento apontado pelo iterador
 - next()
 - recupera o proximo elemento apontado
 - remove()
 - remove o elemento recuperado

Exercícios

- Amplie o programa iniciado na aula anterior, padronizado conforme apresentado anteriormente.
- Além disso, implemente funcionalidade para mostrar o nível da árvore.