



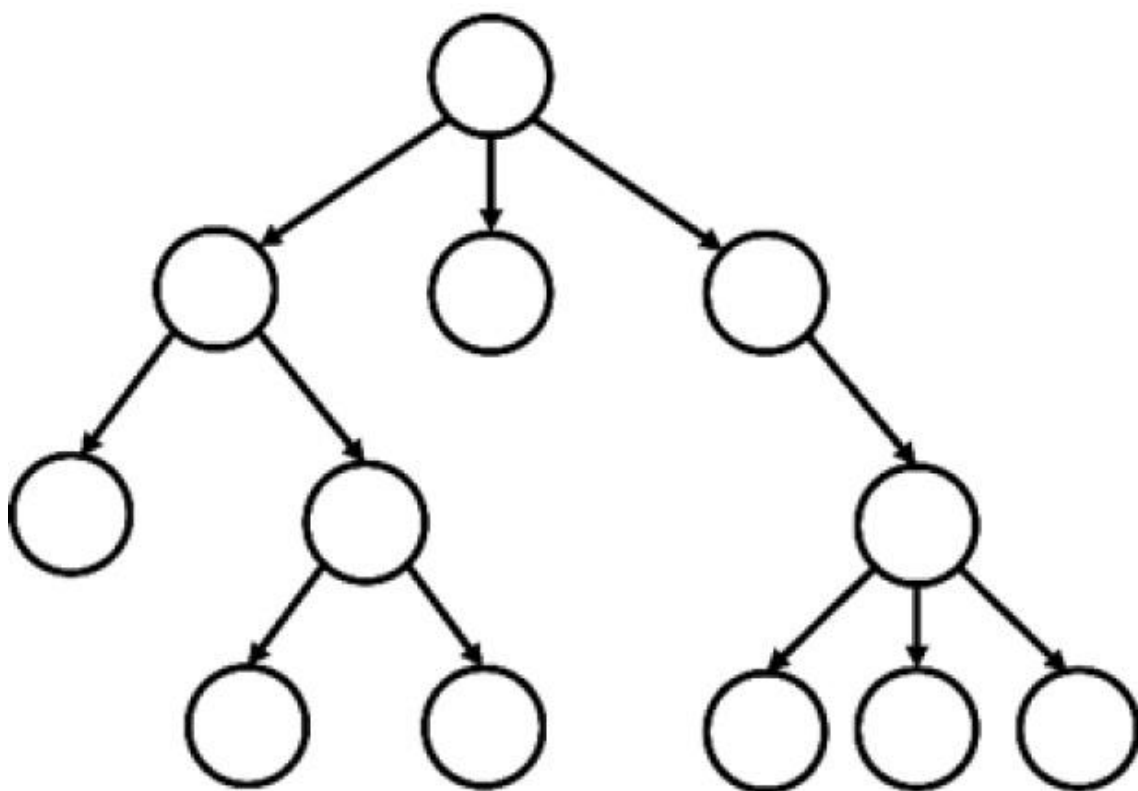
CONCEITOS DE ÁRVORES

Árvores

O que são árvores?

Árvores (*trees*) são estruturas de dados hierárquicas e não linearizadas. Basicamente, as árvores são formadas por um conjunto de elementos, que chamamos de **nós** (ou nodos ou vértices), conectados por um conjunto de arestas. Um dos nós, que dizemos estar no nível 0, é a raiz da árvore e está no topo da hierarquia.

A raiz está conectada a outros nós, no nível 1, que, por sua vez, estão conectados a outros nós, no nível 2, e assim por diante. Os nós que estão no fim da árvore são chamados de **folhas**



As conexões entre os nós de uma árvore seguem uma nomenclatura genealógica. Um nó, em dado nível, está conectado a seus filhos (no nível abaixo) e a seu pai (no nível acima). A raiz da árvore, que está no nível 0, possui filhos, mas não possui pai. Os nós

que estão no final da árvore (os mais distantes da raiz) são chamados de **nós folhas** (*leafs*).

Árvores podem ser desenhadas de muitas formas, porém, na convenção em computação, convém desenhá-las com a raiz no topo, bem diferente das árvores tradicionais que conhecemos na natureza.

1. Nós (*nodes*)

Elemento que contém a informação.

2. Arcos ou arestas (*edges*)

Elementos que ligam dois nós na árvore.

3. Pai (*parents*)

Elemento que atua como o nó superior de um arco.

4. Filho (*child*)

Elemento que atua como nó inferior de um arco.

5. Raiz (*root*)

Elemento que atua como nó que fica no topo da árvore – não tem um nó pai.

6. Folhas (*leafs*)

Elementos que atuam como nós das extremidades inferiores – não têm nós filhos.

7. Grau (*degree*)

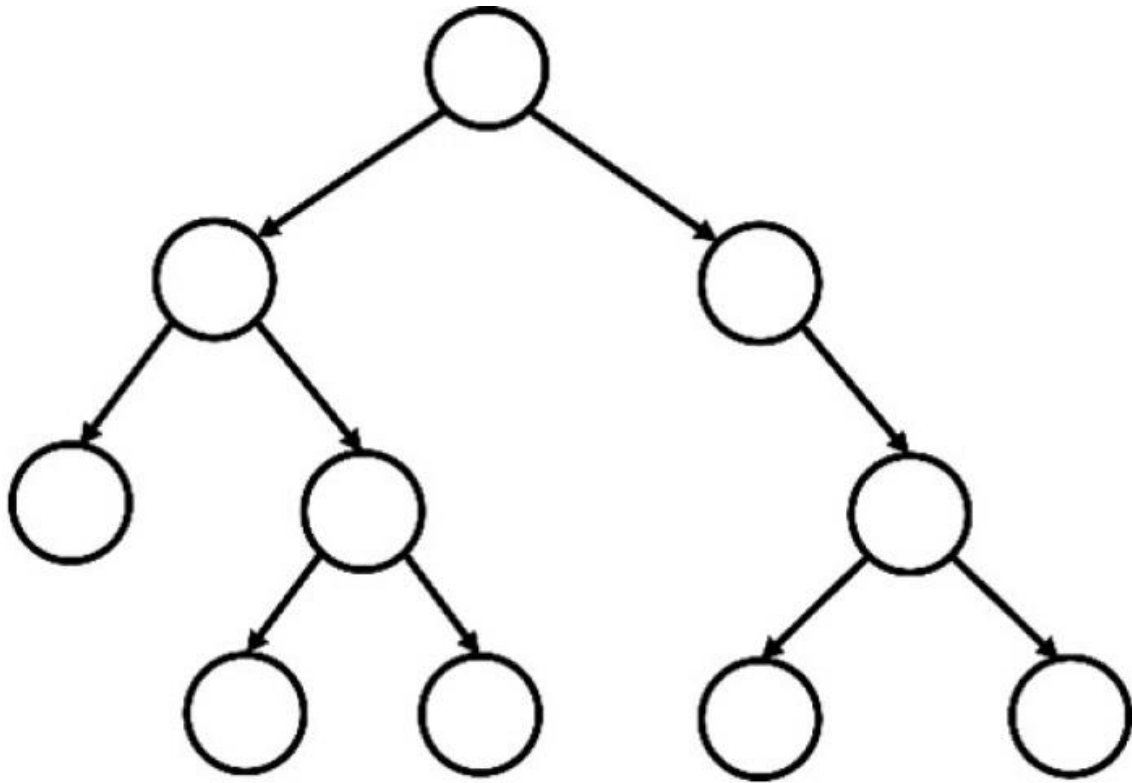
Elemento que representa o número de subárvores de um nó.

Árvores binárias

O que são árvores binárias?

Árvores binárias são árvores nas quais cada nó pode ter, no máximo, dois filhos.

Observe a imagem:

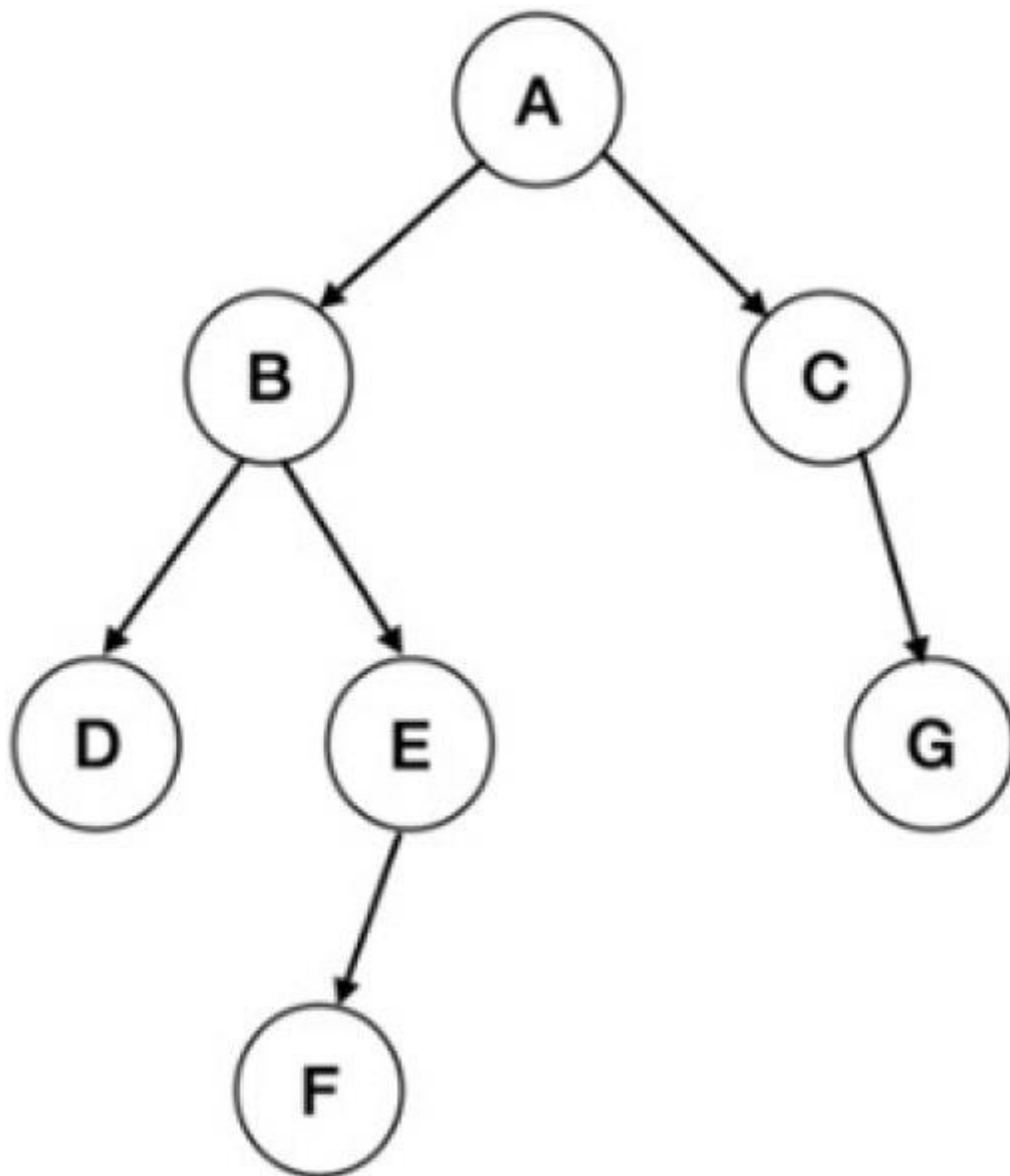


1. A raiz da árvore possui dois filhos, um à direita e outro à esquerda, que, por sua vez, são raízes de duas subárvores.
2. Cada uma dessas subárvores possui uma subárvore esquerda e uma subárvore direita, seguindo esse mesmo raciocínio.

Falando formalmente, uma árvore binária **T** é um conjunto vazio ou é composta pelos seguintes elementos: uma entidade **n**, chamada nó raiz, e pelas entidades **Te** e **Td**, respectivamente, as subárvores esquerda e direita de **T**, que também são árvores binárias.

Observe que, como **Te** e **Td** são árvores binárias, a definição é recursiva, isto é, **Te** possui um nó raiz ou é uma subárvore vazia. O mesmo vale para **Td**.

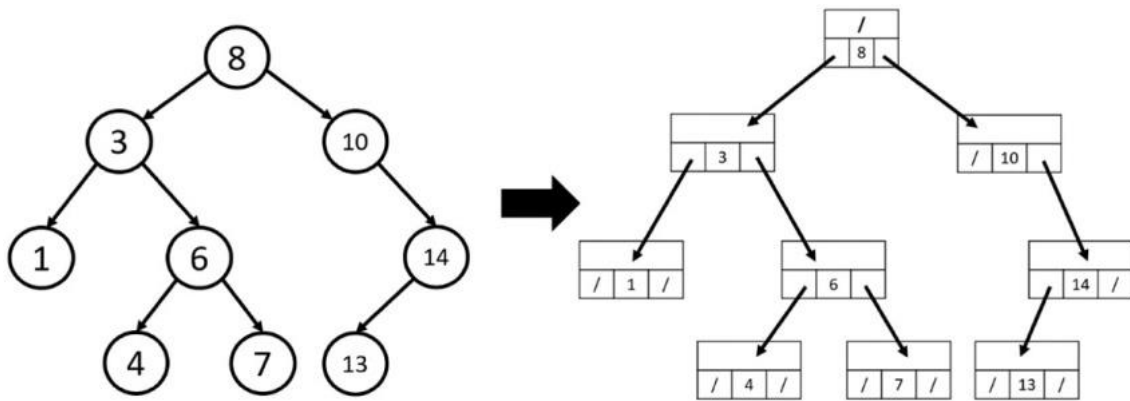
Usualmente, representamos uma árvore por meio da representação gráfica em que se destacam a raiz da árvore **T** e sua subárvore esquerda e direita. Confira a representação:



Na imagem 3, o nó "A" é a raiz da árvore **T**, **Te** é composta pelos nós: "B", "D", "E" e "F"; **Td** é composta pelos nós: "C" e "G". "B" e "C" são as raízes de **Td** e **Te**, respectivamente.

Observe que existe na árvore binária, por definição, distinção entre a subárvore esquerda e direita. Por isso, na representação gráfica, sempre fica evidente a posição esquerda ou direita do nó subordinado à raiz. Na imagem 3, o nó "G" é o filho direito de "C", e não existe, por exemplo, filho esquerdo de "C".

A forma mais comum de representar uma árvore em memória é utilizando alocação dinâmica. Na verdade, não representamos a árvore como um todo, mas uma referência para sua raiz, que guarda a chave (dado) e uma referência para a raiz das subárvores esquerda e direita. Aqui, representaremos nossa árvore usando a linguagem Python.



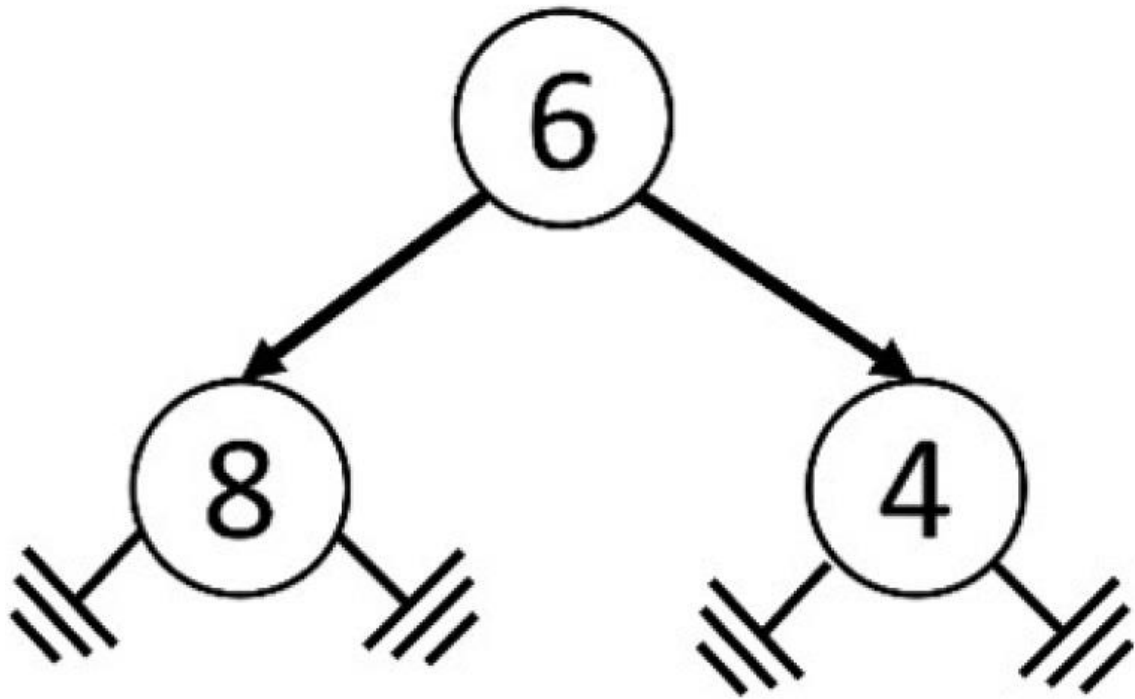
Os nós de uma árvore binária possuem um valor (intitulado chave) e dois apontadores, um para o filho da esquerda e outro para o filho da direita. Esses apontadores representam as ligações (arestas) de uma árvore.

```
class NoArvore:
    def __init__(self, chave = None, esquerda = None, direita = None):
        self.chave = chave
        self.esquerda = esquerda
        self.direita = direita

if __name__ == '__main__':
    raiz = NoArvore(6)
    raiz.esquerda = NoArvore(8)
    raiz.direita = NoArvore(4)
```

A imagem a seguir ilustra a árvore binária implementada no código do algoritmo 1.

Note que o nó raiz (com o valor 6), possui dois filhos, um à esquerda (com o valor 8) e outro à direita (com o valor 4).



Os nós, cujos valores são 8 e 4, não possuem filhos, seus apontadores à esquerda e à direita são **None**, ou seja, não apontam para qualquer outro nó.