

Estrutura de Dados em Python

Prof. Nisston Moraes Tavares de Melo

Estrutura de dados em Python

Dra. Ana Carolina Costa de Oliveira
prof2121@iesp.edu.br

Agenda

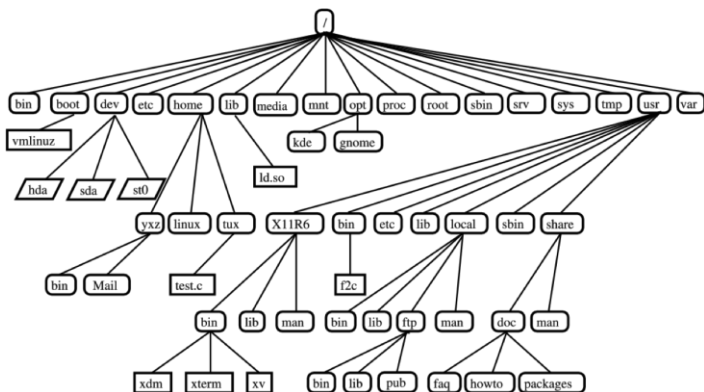
- Árvore binária

1

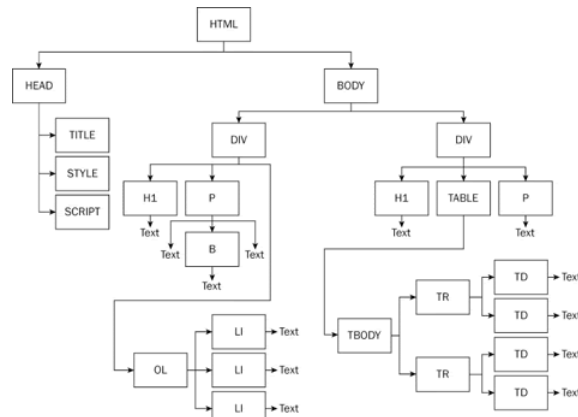
Árvore binária

ÁRVORES

- Uma árvore combina as vantagens de duas estruturas: um vetor ordenado e uma lista encadeada
- Busca rápida (como em um vetor ordenado)
- Inserção e eliminação rápida (como em uma lista encadeada)



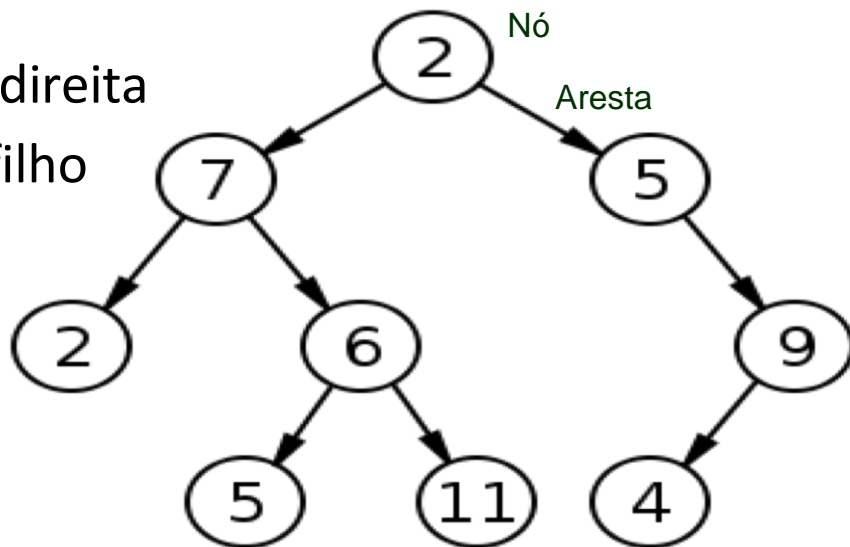
Fonte: <https://www.esli-nux.com/2016/02/aula-5-estrutura-da-arvore-do-sistema.html>



Fonte: <http://devfuria.com.br/javascript/dom/>

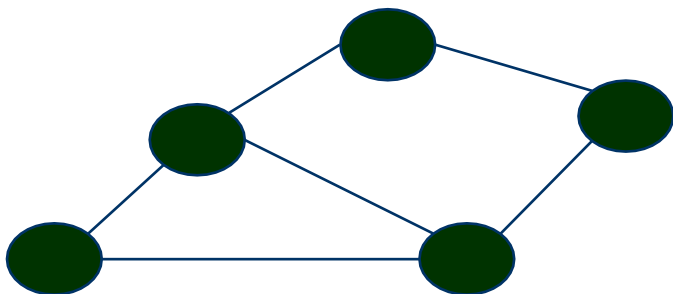
ÁRVORE BINÁRIA

- Uma árvore consiste em nós (círculos) conectados por arestas (linhas)
- Máximo dois filhos
- Filho à esquerda e filho à direita
- Pode ter um ou nenhum filho

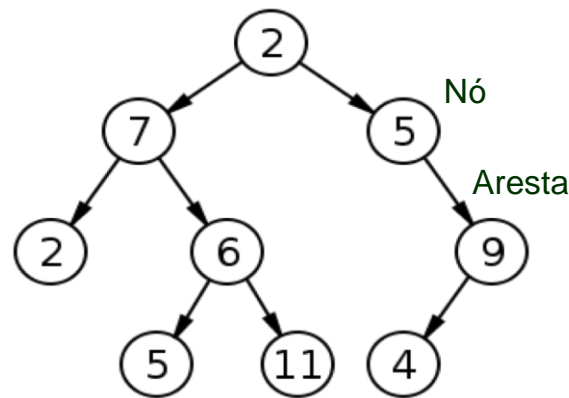


TERMINOLOGIA DE ÁRVORES

- Caminho
 - Caminho que liga um nó até outro nó
- Raiz
 - É o nó na parte superior. Há apenas uma raiz em uma árvore e deve haver somente um caminho da raiz até qualquer outro nó

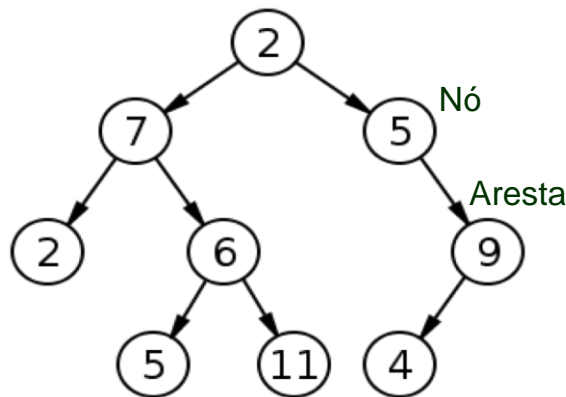


Não-árvore



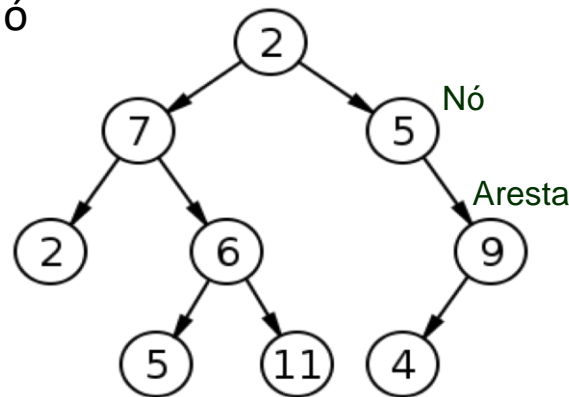
TERMINOLOGIA DE ÁRVORES

- Pai
 - Qualquer nó (exceto a raiz) tem exatamente uma aresta que sobe para outro nó. O nó acima dele é chamado de pai do nó
- Filho
 - Qualquer nó pode ter uma ou mais linhas descendo para outros nós. Esses nós abaixo de um dado nó são chamados de seus filhos
- Folha
 - Um nó que não tem filhos



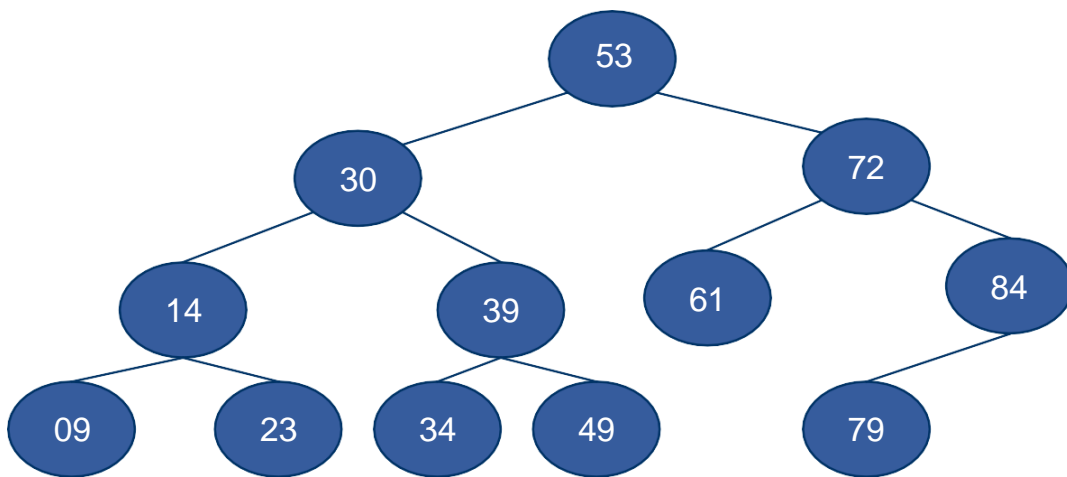
TERMINOLOGIA DE ÁRVORES

- Subárvore
 - Qualquer nó pode ser considerado como sendo a raiz de uma subárvore, que consiste em seus filhos
- Visitando
 - Um nó é visitado quando o controle do programa chega ao nó, em geral para a finalidade de executar alguma operação do nó
- Percorrendo
 - Visitar todos os nós em alguma ordem especificada
- Níveis
 - O nível de um determinado nó refere-se a quantas gerações o nó está da raiz
- Chaves
 - Valor usado para buscar um item



ÁRVORE BINÁRIA DE BUSCA

- O filho à esquerda de um nó tem que ter uma chave menor que seu pai e o filho à direita de um nó tem que ter uma chave maior ou igual ao seu pai



ÁRVORE BINÁRIA DE BUSCA – INSERÇÃO

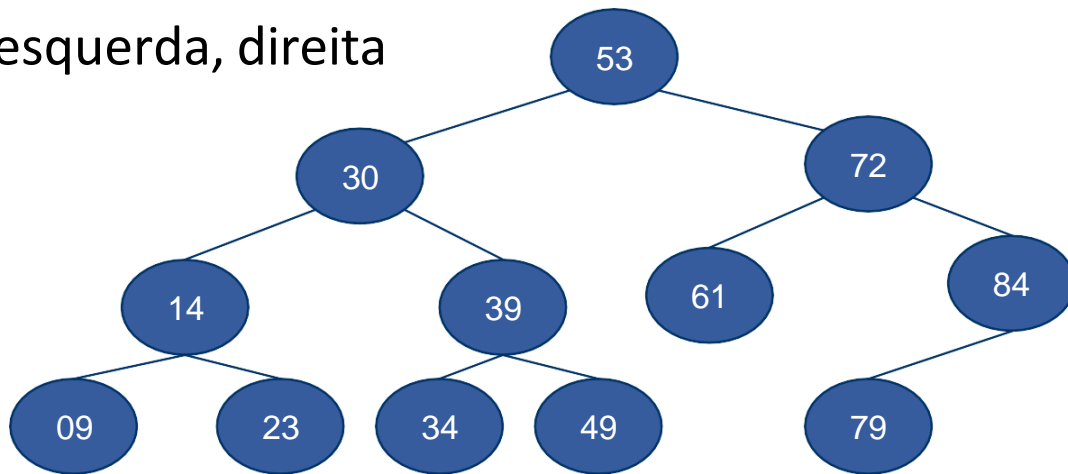
- Primeiro, o local para inserir deve ser encontrado
- Segue-se o caminho da raiz até o devido nó, que será pai do novo nó
- Quando esse pai for localizado, o novo nó será conectado como seu filho à esquerda ou a direita, dependendo da chave do novo nó ser menor ou maior que a do pai
- Visualização on-line: <https://visualgo.net/en/bst>
- Big-O: $O(\log n)$ para o caso médio e $O(n)$ no pior caso

ÁRVORE BINÁRIA DE BUSCA – PESQUISA

- Procurar nas subárvores da esquerda ou direita
- Visualização on-line: <https://visualgo.net/en/bst>
- Big-O: $O(\log n)$ para o caso médio e $O(n)$ no pior caso

ÁRVORE BINÁRIA – TRAVESSIA PRÉ-ORDEM

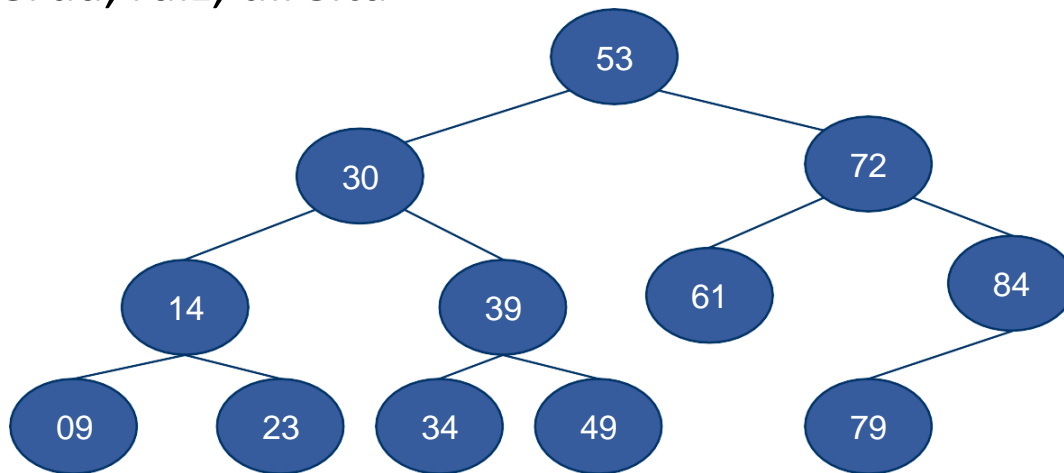
- Primeiro visita a raiz e depois recursivamente faz uma travessia na subárvore esquerda, seguido de uma travessia recursiva na subárvore direita
- Raiz, esquerda, direita



- 53
- 30
- 14
- 9
- 23
- 39
- 34
- 49
- 72
- 61
- 84
- 79

ÁRVORE BINÁRIA – TRAVESSIA EM ORDEM

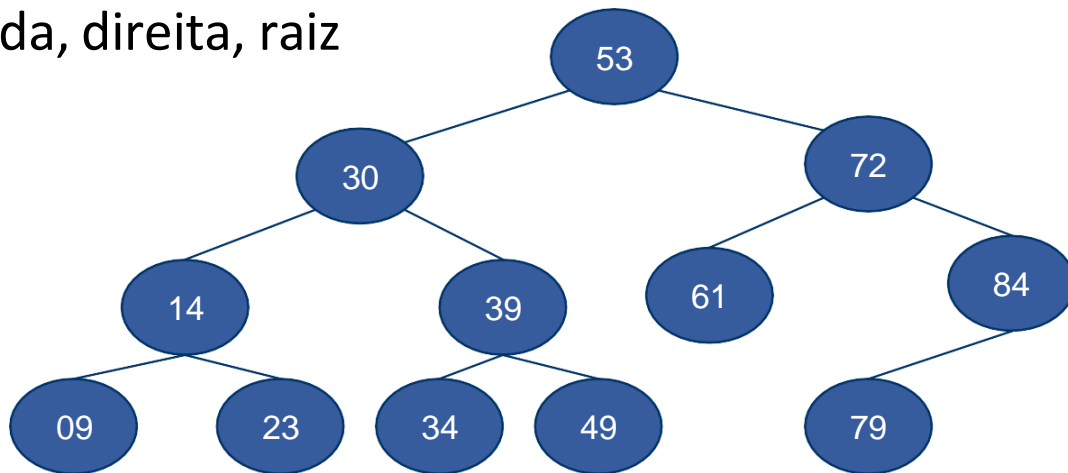
- Recursivamente faz a travessia na subárvore esquerda, visita a raiz e faz uma travessia recursiva na subárvore direita
- Esquerda, raiz, direita



- 09
- 14
- 23
- 30
- 34
- 39
- 49
- 53
- 61
- 72
- 79
- 84

ÁRVORE BINÁRIA – TRAVESSIA PÓS-ORDEM

- Recursivamente faz a travessia na subárvore esquerda, faz uma travessia recursiva na subárvore direita e por fim visita a raiz
- Esquerda, direita, raiz



- 09
- 23
- 14
- 34
- 49
- 39
- 30
- 61
- 79
- 84
- 72
- 53



Exercício.



uniesp

Centro Universitário