

ÁRVORES BINÁRIAS

DCE792 - AEDs II (Prática)

Atualizado em: 23 de setembro de 2025

Iago Carvalho

Departamento de Ciência da Computação



Uma árvore é uma estrutura comum em computação utilizada na modelagem e resolução de diversos problemas diferentes

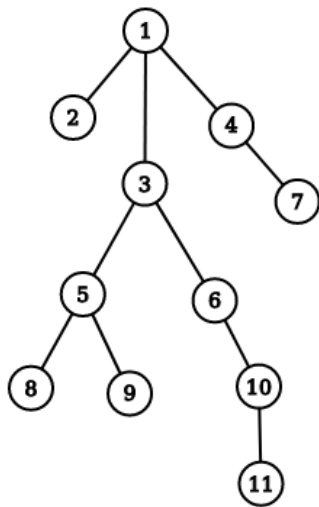
Muito comum em grafos

- Seja $G = (N, E)$ um grafo
- N é um conjunto de vértices
- E é um conjunto de arestas

Neste sentido, uma árvore é um subconjunto $E' \subseteq E$ tal que

- Estrutura conexa
- Não existem ciclos
 - Existem menos que N arestas

ÁRVORE EM UM GRAFO



ÁRVORES COMO ESTRUTURAS DE DADOS

Árvores em grafos serão estudadas em AEDS III. Nesta disciplina estamos interessados em árvores quanto estruturas de dados

- Estruturas não lineares
- Implementadas com apontadores
- Operações, no geral, recursivas
- Utilizam o conceito de vértices e ponteiros
- Vértice pai e vértice filho

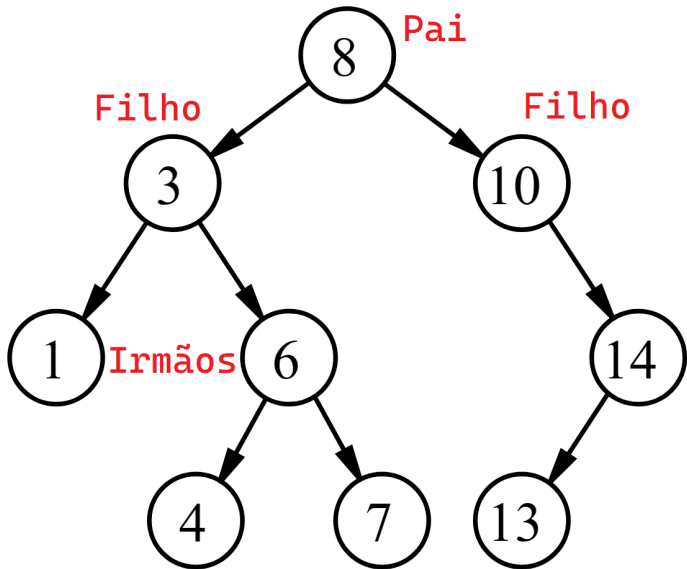
Árvores são úteis para armazenar dados de forma ordenada

- Operação de busca é executada em tempo logaritmico

Inicialmente, estamos interessados em árvores binárias

- Uma árvore onde cada pai tem, no máximo, dois filhos

ÁRVORES BINÁRIAS



REPRESENTAÇÃO DE UMA ÁRVORE BINÁRIA

```
1  struct node {  
2      int valor;  
3      struct node *esquerda, *direita;  
4  };
```

valor: Elemento da árvore

esquerda: Apontador para o filho a esquerda

direita: Apontador para o filho a direita

OPERAÇÕES POSSÍVEIS COM ÁRVORES BINÁRIAS

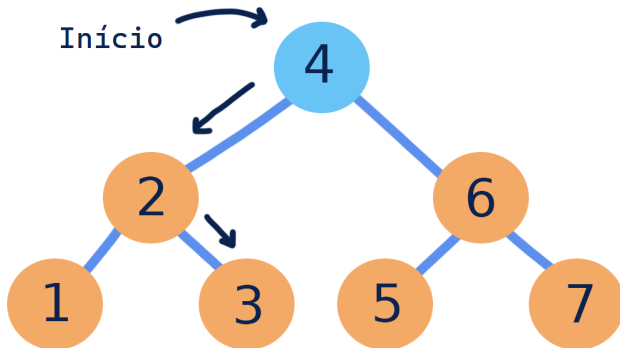
- `struct node* novo_no(<valor>);`
 - Cria um novo nó na árvore
- `struct node* insere(<valor>);`
 - Insere um novo item na árvore
- `struct node* buscar(<valor>);`
 - Verifica se existe um determinado item na árvore

FUNÇÃO BUSCAR

Sabe-se que o filho a esquerda possui valor menor que o pai, enquanto o filho a direita possui valor maior que o pai

Esta estrutura é utilizada no processo de busca

Busca pelo valor 3

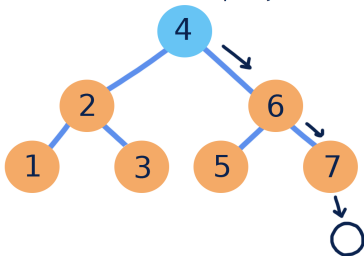


FUNÇÃO INSERIR

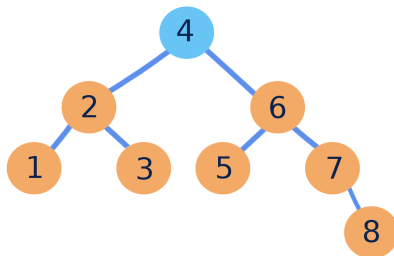
Busca pelo valor. Caso ele não exista, faz a inserção

Inserir o valor 8

1. encontrar a posição



2. Criar o novo nó



O código completo de uma árvore binária está disponível no Github



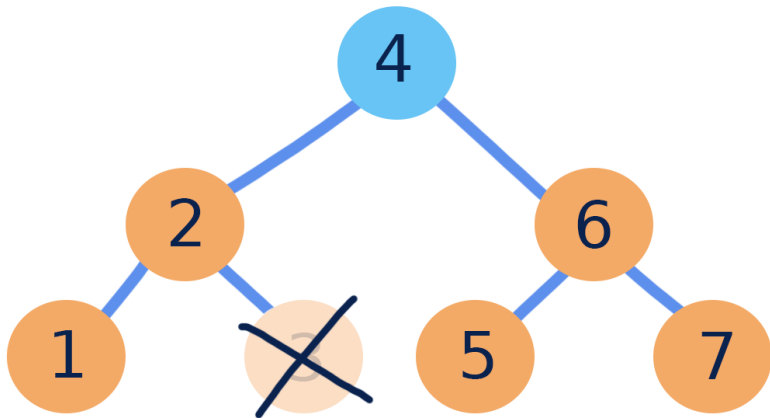
► Link

ATIVIDADE PRÁTICA

ATIVIDADE

Modificar o código do Github para implementar a operação de remoção de nós

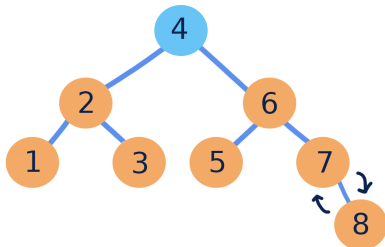
Remover o valor 3



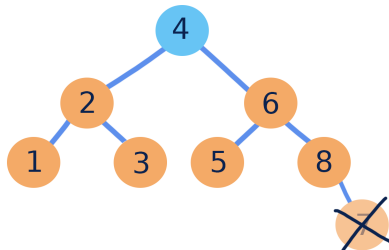
A operação para remover nós com um filho é diferente

Remover itens com um filho

1. Troque o nó com seu filho

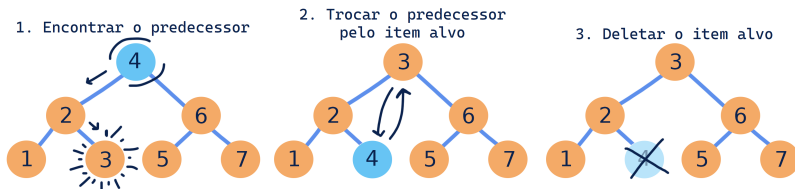


2. Faça a remoção



Com dois filhos, é mais diferente ainda

Remover items com dois filhos



PRÓXIMA AULA:

CAMINHAMENTO EM
PRÉ-ORDEM,
ORDEM, E
PÓS-ORDEM