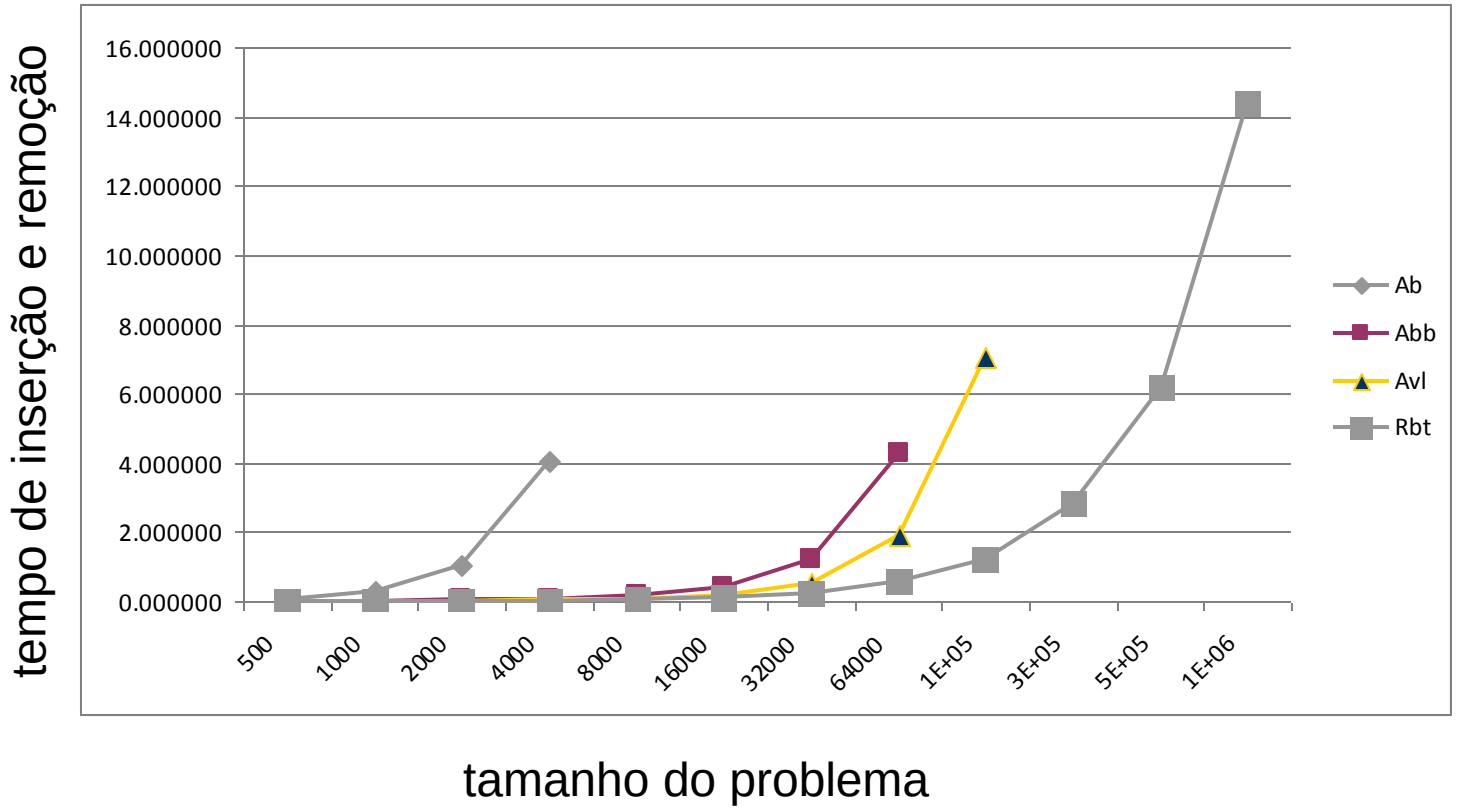
Estrutura de Dados 2

Árvore Binária

Prof. Rafael Oliveira Vasconcelos

O que vamos estudar de árvores binárias?

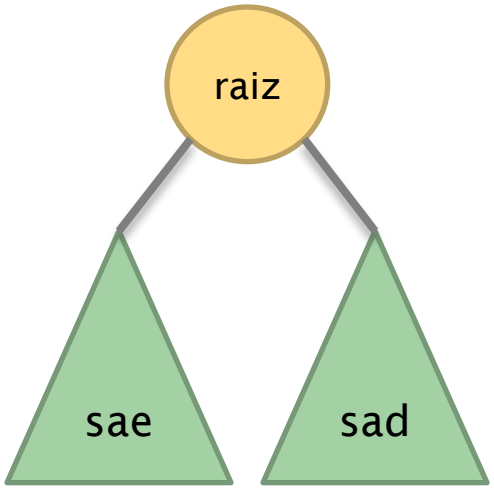
Ab = árvore binária
Abb = árvore binária de busca
Avl = árvore binária de busca, balanceada
Rbt = árvores rubro-negras ou vermelho e preto (red-black tree)



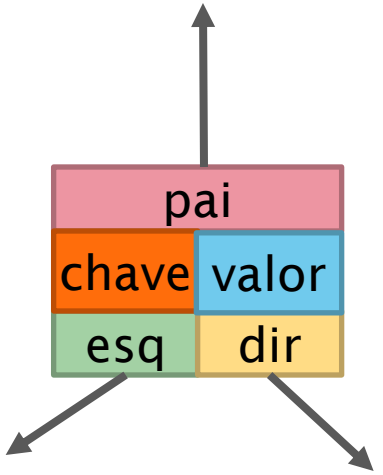
Árvore Binária (AB) - Definições

É uma árvore com as seguintes propriedades:

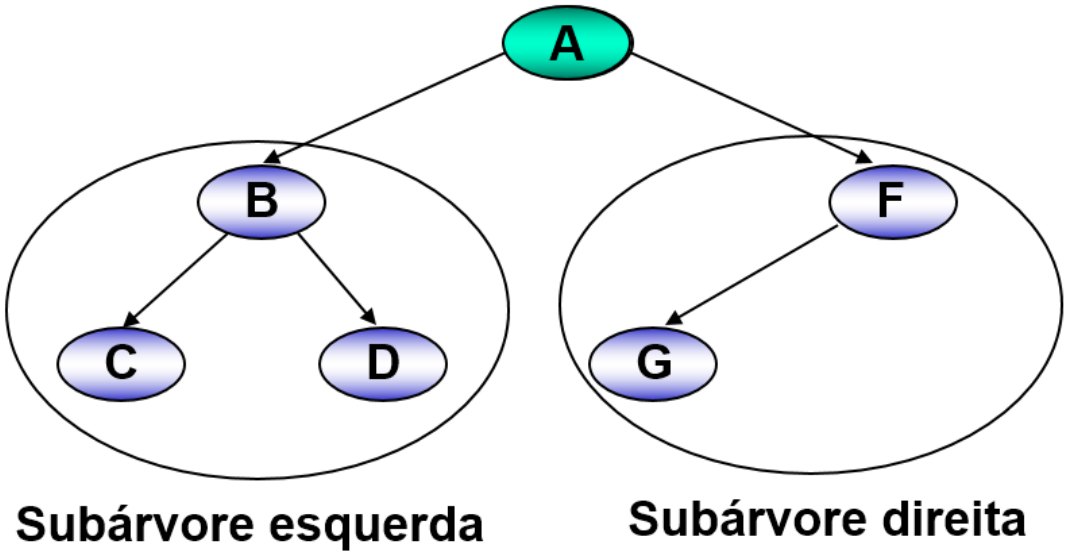
- Cada nó possui no máximo 2 subárvores (grau 2)
- Cada subárvore é identificada como subárvore esquerda ou direita

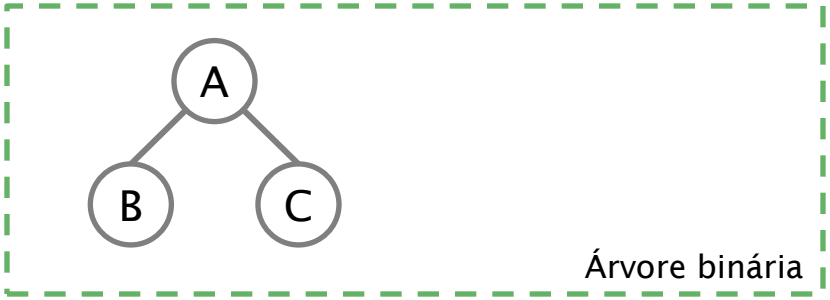


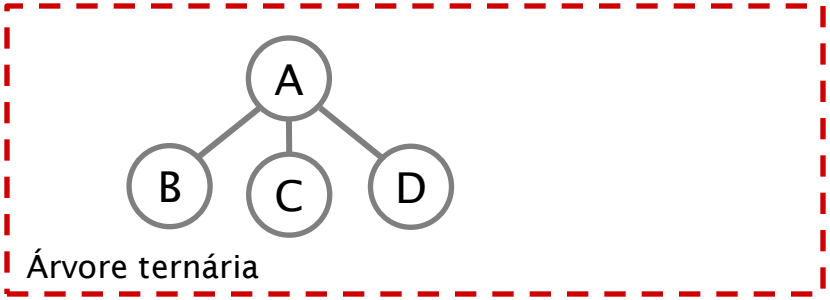

```
class NoArvoreBin{
    Comparable obj;
    NoArvoreBin esq;
    NoArvoreBin dir;
}
```



Árvore Binária - Exemplo

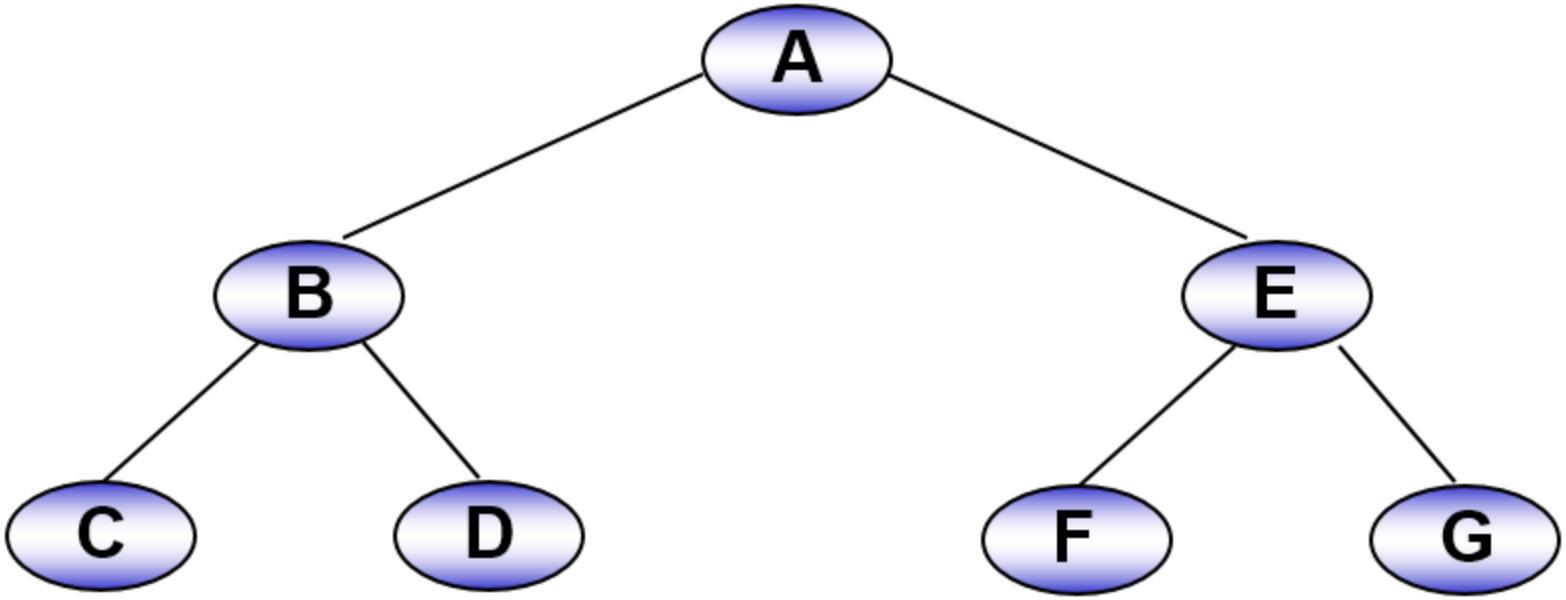






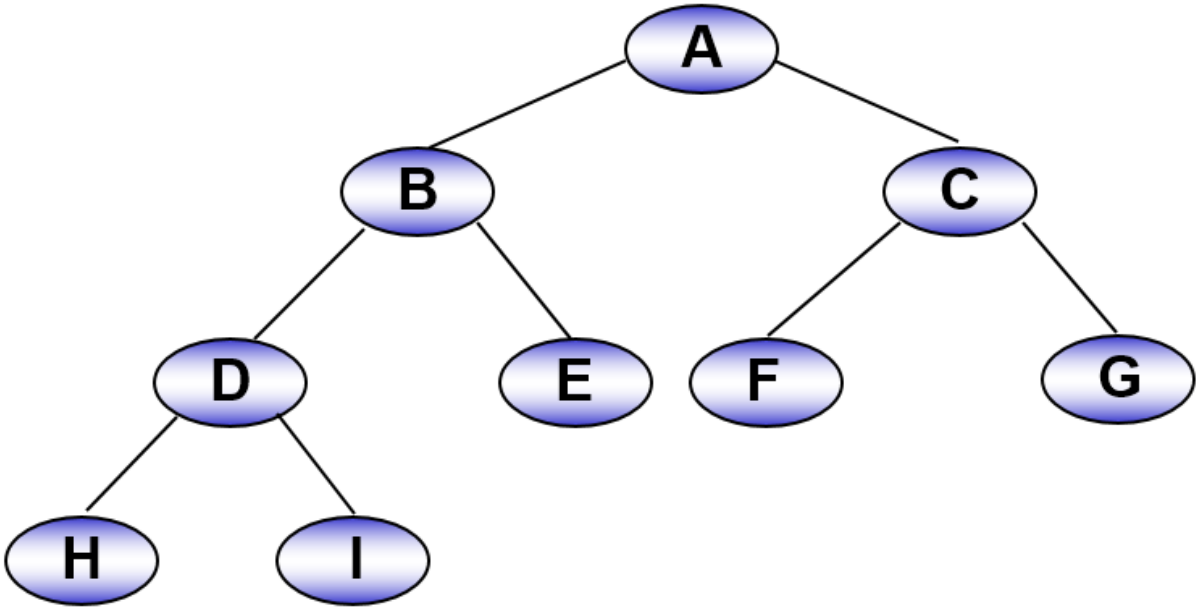
Árvore Binária Cheia

- Uma árvore binária é denominada cheia se:
  - Todas as folhas estão no mesmo nível
  - Todos os nós não-folhas têm exatamente 2 filhos



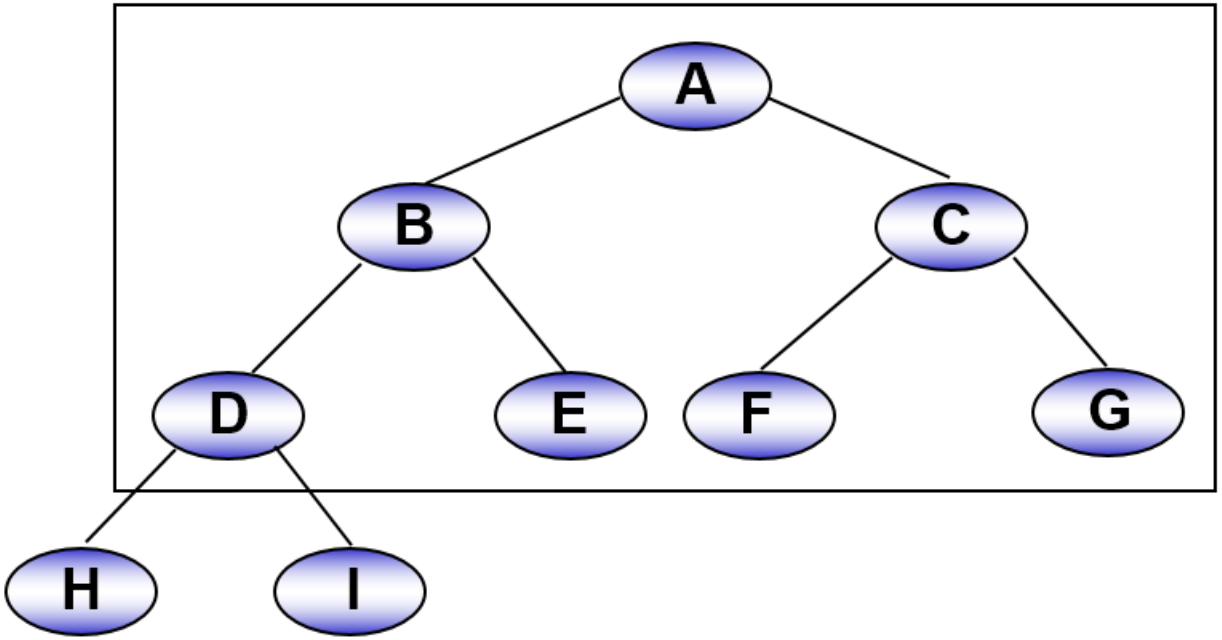
Árvore Binária Completa

- Uma árvore binária é denominada completa se:
  - Ela for cheia até a altura h-1
  - Todos os nós do nível h ocupam as posições mais a esquerda da árvore



Árvore Binária Completa

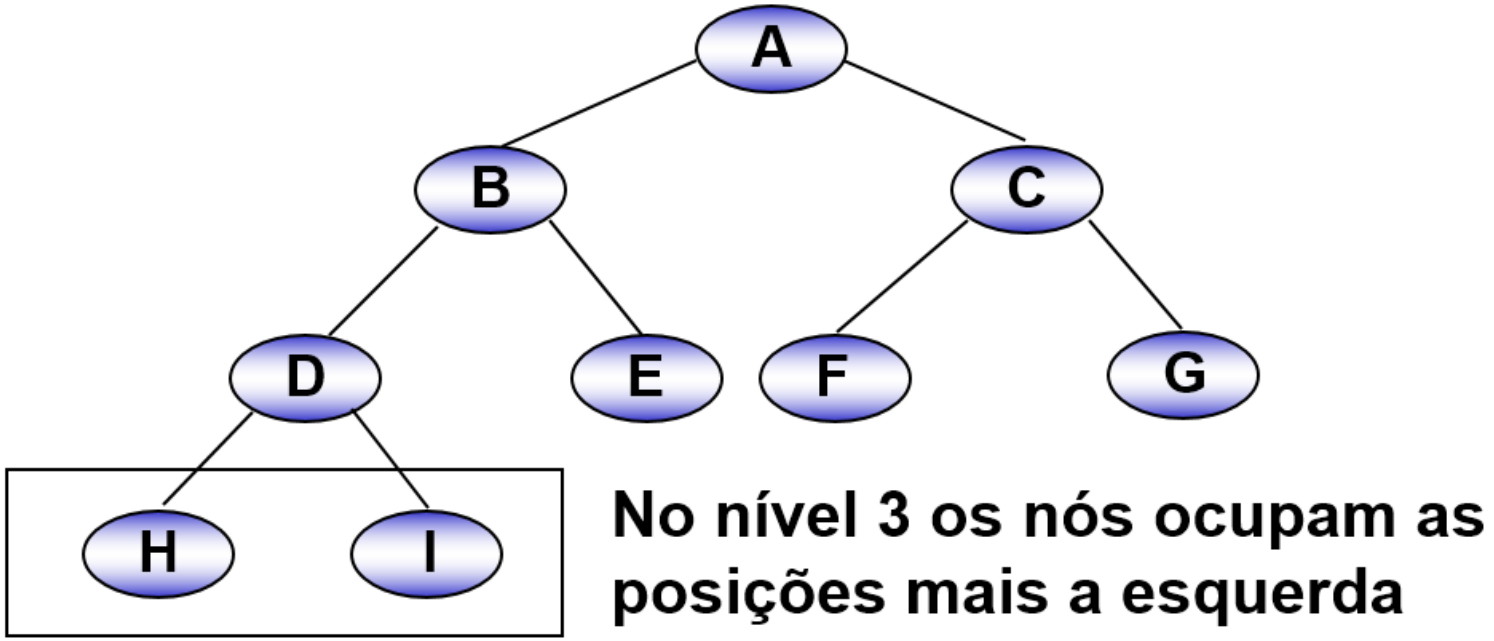
- Uma árvore binária é denominada completa se:
  - Ela for cheia até a altura h-1
  - Todos os nós do nível h ocupam as posições mais a esquerda da árvore



Cheia até o nível 2

Árvore Binária Completa

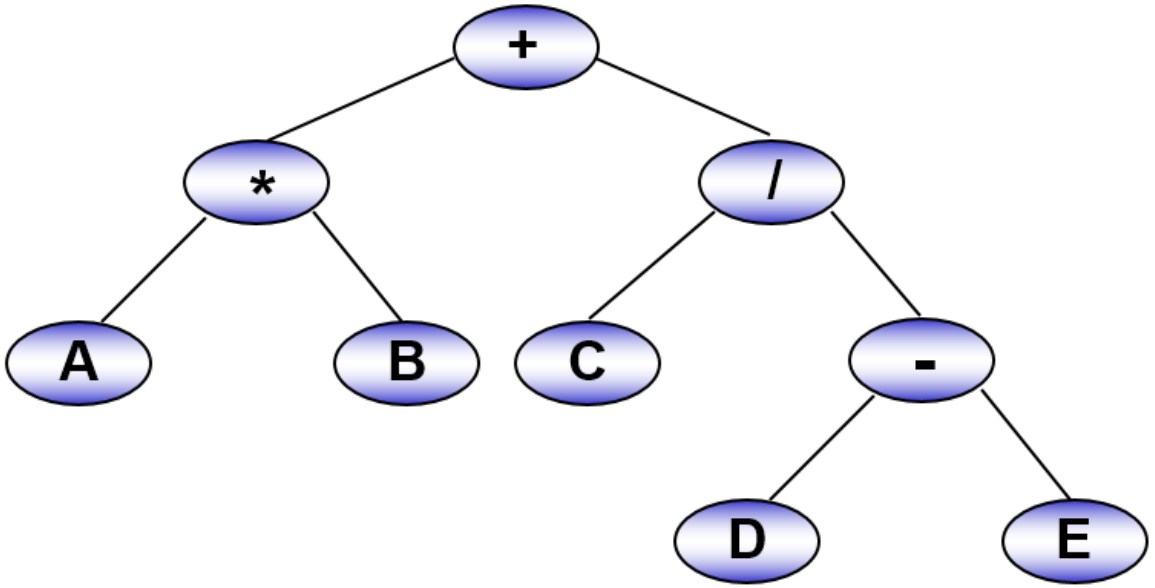
- Uma árvore binária é denominada completa se:
  - Ela for cheia até a altura h-1
  - Todos os nós do nível h ocupam as posições mais a esquerda da árvore



Representação de uma expressão aritmética em uma Árvore Binária

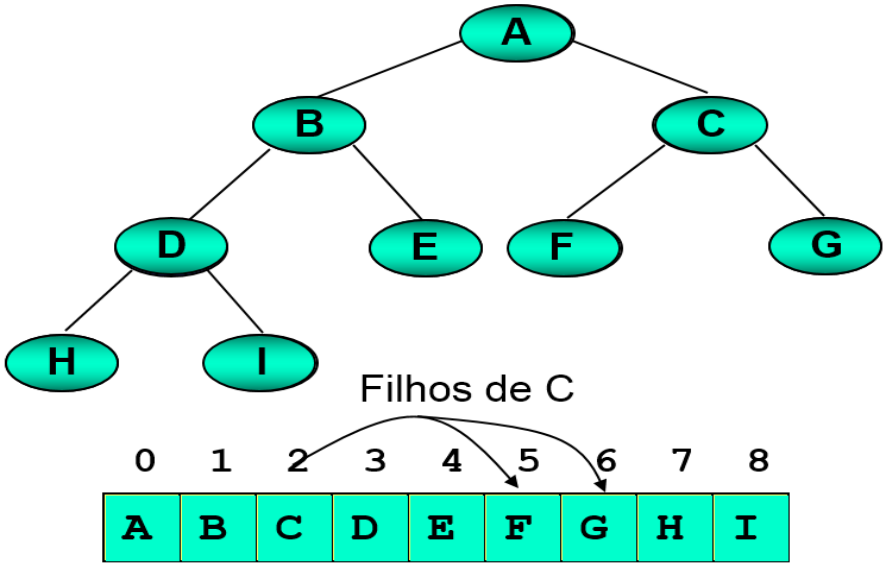
- Cada operador é armazenado em um nó da árvore
- Suas subárvores são os operandos

A * B + C / (D - E)



Implementação de Árvores Binárias

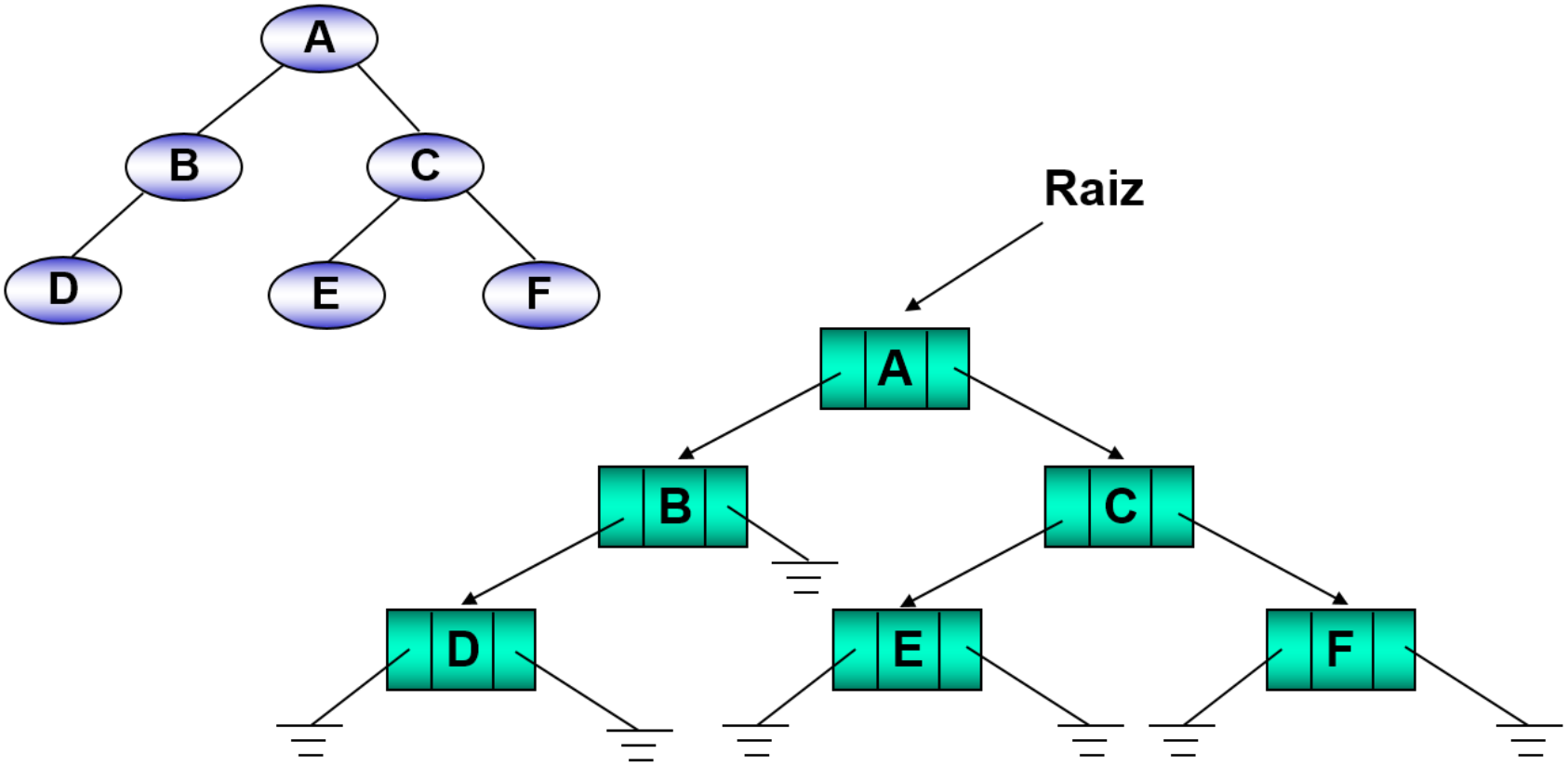
- Implementação usando vetor é conveniente apenas quando a árvore for completa



- Para um nó que se encontra na posição i
  - Seu pai estará na posição (i + 1) / 2 - 1
  - Seus filhos estarão nas posições 2 * (i + 1) - 1 e 2 * (i + 1)

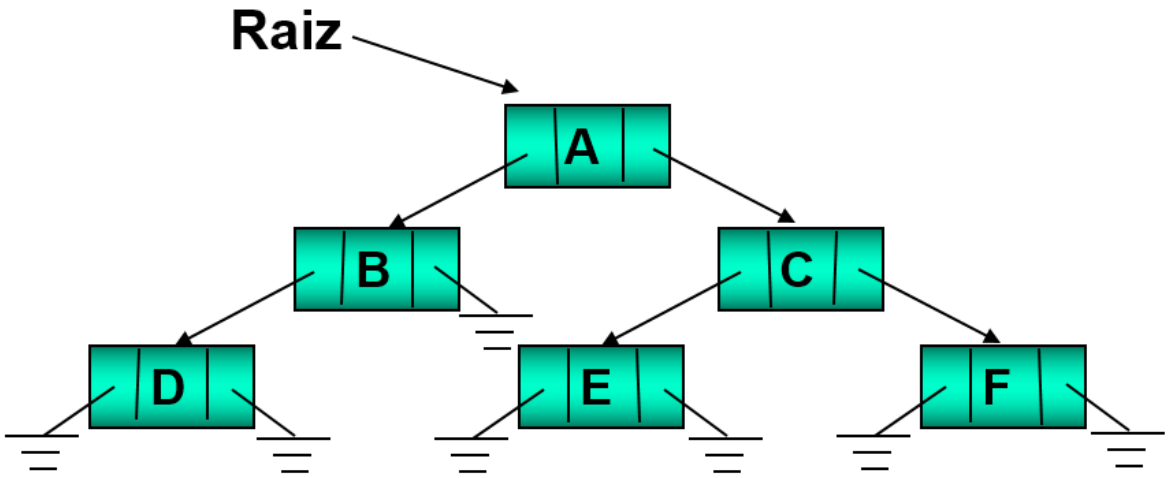
Implementação de Árvores Binárias

- Implementação usando ponteiros



Implementação de Árvores Binárias

- Implementação usando ponteiros




```
class NoArvoreBin{
    Comparable obj;
    NoArvoreBin esq;
    NoArvoreBin dir;
}
```

Dúvidas?

Exercício 1 - Árvore

- Implemente uma árvore binária usando vetor com as seguintes operações
  - Inserção
  - Consulta
  - Remoção
  - Navegação
- Como deverá ser a classe Nó?
- Interface sugerida para a árvore
  - No<Chave, Valor> inserir(Chave chave, Valor valor, No<T> pai, Lado lado);
  - No<Chave, Valor> inserir(Chave chave, Valor valor, int indice);
  - No<Chave, Valor> inserir(Chave chave, Valor valor);
  - No<Chave, Valor> obterFilho(No<Chave, Valor> pai, Lado lado);
  - No<Chave, Valor> obterFilho(Chave chave, Lado lado);
  - No<Chave, Valor> obterNo(int indice);
  - No<Chave, Valor> remover(No<Chave, Valor> no);
  - No<Chave, Valor> remover(No<Chave, Valor> pai, Lado lado);
  - No<Chave, Valor> remover(int indice);
  - No<Chave, Valor> remover(Chave chave);
  - Collection<Valor> obterValores();
- Lembre de fazer as verificações devidas

Exercício 2

- Implemente um programa que utilize a árvore criada no exercício anterior para armazenar cidades

- Uma cidade possui
  - Um ID numérico
  - Um nome

- O programa deve
  - Permitir a inclusão, remoção e consulta das cidades
  - Listar todas as cidades cadastradas