



# Aula 13-05-2024- Árvore Binária

## Inserção

- 1- Se a raiz estiver vazia, insere-se o elemento nela
- 2- Senão, se o novo elemento for menor que a raiz, inserir na esquerda
- 3- Senão, se o novo elemento for maior que a raiz, inserir na direita

## Inserção em Java com Retorno de Referencia

- A passagem de parametros em Java e em C é sempre por VALOR

## Inserção em Java com passagem de pai

- Melhor caso Teta(1) : inserindo na raiz
- Pior caso Teta(n) : ordem crescente ou decrescente
- Caso medio Teta(log(n)): quando inserimos na folha de uma arvore balanceada

## Caminhamento

- Percorrer a árvore inteira

- Somar todos os elementos:

```
public int soma(No i){
    int resp = 0;
    if(i!= null){
        resp = i.elemento + soma(i.esq)+soma(i.dir);
    }
    return resp;
}
```

- Somar todos os elementos IMPARES

```
public int somaImp(No i){
    int resp = 0;
    if(i!= null){
        if(i.elemento %2 != 0){
            resp += i.elemento;
        }
        resp += somaImp(i.esq)+somaImp(i.dir);
    }
}
```

- Somar todos os elementos PARES

```
public int somaPar(No i){
    int resp = 0;
    if(i!=null){
        resp = (i.elemento % 2 == 0)? 1:0 + somaPar(i.esq)+somaPar(i.dir);
    }
    return resp;
}
```

- Retornar true se a árvore conter algum número divisível por 11

```
public boolean hasDiv11(No i){
    boolean resp = false;
    if( i !=null){
        resp = (i.elemento %11 == 0) || hasDiv11(i.esq) || hasDiv11(i.dir);
    }
}
```

```
}  
    return resp;  
}
```

## Estruturas Híbridas

**ENTREGAR PARA A PROXIMA AULA**

### Unidade 05g

- Faça TODOS os exercícios no papel