ÁRVORE BINÁRIA

Alunos: João Victor, Katrina, Thacio e Vitória
Professor: Jocivan Suassone Alves

UNITINS

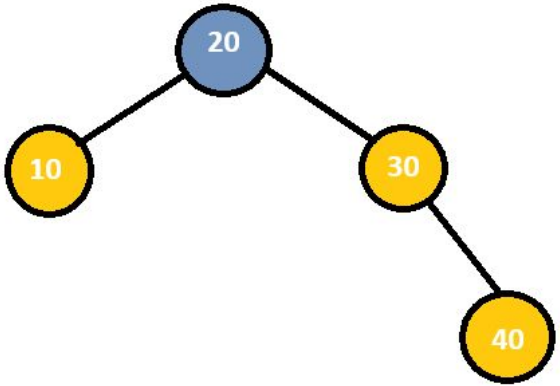
GOVERNO DO TOCANTINS
TRABALHANDO E CUIDANDO DE TODOS

ALTURA DA ÁRVORE BINÁRIA

A função que mede a altura inteira da árvore, é medida a partir do nó raiz, começando a partir de 0.

```
public int alturaArvore(No noArvore){
        if(noArvore == null){
            return -1;
        }
        else{
            int ae = alturaArvore(noArvore.esquerda);
            int ad = alturaArvore(noArvore.direita);

            if(ae < ad){
                return ad + 1;
            }else{
                return ae + 1;
            }
        }
    }
```



UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

FATOR DE BALANCEAMENTO DA ÁRVORE BINÁRIA

Para que uma árvore binária esteja balanceada, seu resultado do fator de balanceamento deve estar entre (-1, 0, 1).

```
Fator de balanceamento = altura esquerda - altura direita => (-1, 0, 1)
```

```java
public int fatorBalanceamento(No noArvore) {
        if (noArvore == null) {
            return 0;
        }
        int fatorBalanceamento = alturaArvore(noArvore.esquerda) - alturaArvore(noArvore.direita);

        if (fatorBalanceamento == 1 || fatorBalanceamento == 0 || fatorBalanceamento == -1) {
            System.out.println("Árvore balanceada.");
            return fatorBalanceamento;
        } else {
            System.out.println("Árvore desbalanceada.");
            return fatorBalanceamento;
        }
    }
```

UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

ÁRVORE AVL

- Tipos de Rotação
  - Rotação RR
  - Rotação LL
  - Rotação RL
  - Rotação LR

ROTAÇÃO À ESQUERDA

A rotação à esquerda é aplicada quando há um desbalanceamento à esquerda em um nó, ou seja, o filho esquerdo tem uma altura maior que a altura do filho direito. Isso geralmente ocorre quando uma nova chave é inserida no subárvore à direita do filho esquerdo.

```
private No rotacaoEsquerda(No y) {
        No x = y.direita;
        No esquerdaDoAtual = x.esquerda;

        x.esquerda = y;
        y.direita = esquerdaDoAtual;

        atualizarAltura(y);
        atualizarAltura(x);

        return x;
        }
```

y: O nó desbalanceado que precisa ser rotacionado.

x: O novo nó raiz após a rotação.

esquerdaDoAtual: A subárvore que precisa ser movida da esquerda de x para a direita de y.

UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

ROTAÇÃO À DIREITA

A rotação à direita é aplicada quando há um desbalanceamento à direita em um nó, ou seja, o filho direito tem uma altura maior que a altura do filho esquerdo. Isso geralmente ocorre quando uma nova chave é inserida no subárvore à esquerda do filho direito.

```
private No rotacaoDireita(No x) {
        No y = x.esquerda;
        No direitaDoAtual = y.direita;

        y.direita = x;
        x.esquerda = direitaDoAtual;

        atualizarAltura(x);
        atualizarAltura(y);

        return y;
    }
```

x: O nó desbalanceado que precisa ser rotacionado.

y: O novo nó raiz após a rotação.

direitaDoAtual: A subárvore que precisa ser movida da direita de y para a esquerda de x.

UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

BALANCEANDO

```
// Calcular o fator de balanceamento do nó
int fatorBalanceamento = alturaNo(no.esquerda) - alturaNo(no.direita);

// Caso do fator de balanceamento positivo (desbalanceamento à esquerda)
if (fatorBalanceamento > 1) {
    // Caso LL: Rotação à direita
    if (alturaNo(no.esquerda.esquerda) >= alturaNo(no.esquerda.direita)) {
        return rotacaoDireita(no);
    } else { // Caso LR: Rotação à esquerda e depois à direita
        no.esquerda = rotacaoEsquerda(no.esquerda);
        return rotacaoDireita(no);
    }
}
// Caso do fator de balanceamento negativo (desbalanceamento à direita)
else if (fatorBalanceamento < -1) {
    // Caso RR: Rotação à esquerda
    if (alturaNo(no.direita.direita) >= alturaNo(no.direita.esquerda)) {
        return rotacaoEsquerda(no);
    } else { // Caso RL: Rotação à direita e depois à esquerda
        no.direita = rotacaoDireita(no.direita);
        return rotacaoEsquerda(no);
    }
}
return no;
```

- Cálculo do Fator de Balanceamento;
- Verificações e Rotações;
- Retorno do Nó Balanceado;

UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

IMPLEMENTAÇÃO DA ÁRVORE AVL

```java
public void inserir(int elemento) {
    this.raiz = inserir(this.raiz, elemento);
}

private No inserir(No no, int elemento) {
    if (no == null) {
        return new No(elemento);
    }

    if (elemento < no.elemento) {
        no.esquerda = inserir(no.esquerda, elemento);
    } else if (elemento > no.elemento) {
        no.direita = inserir(no.direita, elemento);
    } else {
        // Elemento já existe
        return no;
    }

    atualizarAltura(no);

    return balancear(no);
}
```

Inserir Nó

- Verificar se o nó raiz existe;
- Identificar o elemento inserido para qual o ramo da árvore;
- Caso o elemento seja menor que o elemento do nó raiz, insere na esquerda, caso seja maior, insere na direita.

UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

IMPLEMENTAÇÃO DA ÁRVORE AVL

```java
public No buscar(int elemento) {
    return buscar(raiz, elemento);
}

private No buscar(No no, int elemento) {
    if (no == null || no.elemento == elemento) {
        return no;
    }

    if (elemento < no.elemento) {
        return buscar(no.esquerda, elemento);
    } else {
        return buscar(no.direita, elemento);
    }
}
```

Buscar Nó

Na busca do nó, é feito a comparação a partir da raiz com elemento da esquerda, se for menor, ou da direita, se for maior.

UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS


UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS