INSTITUTO DE INFORMÁTICA
Universidade Federal de Goiás

Dicionário em Árvore: Uma abordagem AVL

Arthur Teixeira Perillo
Arthur Trindade da Silva
João Felipe Carlos Rodrigues
Kevin Brunno da Cunha Oliveira
Rubens Augusto Medeiros Miranda

INSTITUTO DE INFORMÁTICA
UFG

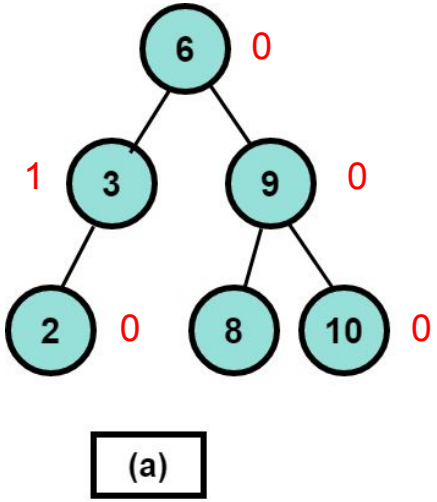
UFG
UNIVERSIDADE FEDERAL DE GOIÁS

O que vamos apresentar?

Mas o que é uma árvore balanceada?

Balanceamento AVL

1. Uma árvore binária balanceada (AVL) é uma árvore binária na qual as alturas das duas subárvores de todo nó nunca diferem em mais de 1.
2. A transformação a ser feita na árvore tal que ela se mantenha balanceada é chamada de rotação



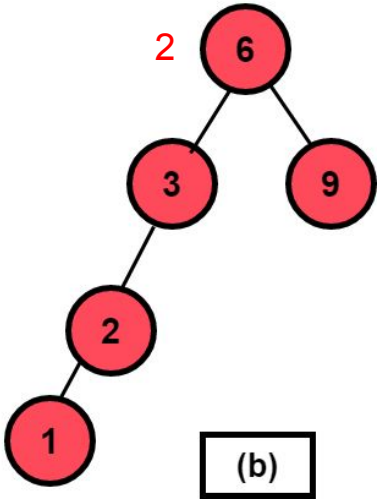


Tabela da AVL

Tabela 1: Complexidade referente a árvore AVL

	Média	Pior Caso
Altura	*O(n)*	*O(n)*
Busca	*O(log n)*	*O(log n)*
Inserção	*O(log n)*	*O(log n)*
Exclusão	*O(log n)*	*O(log n)*

Função para realizar uma rotação simples à direita

```
struct Node* rotateRight(struct Node* z)
    struct Node* y = z->left;
    struct Node* T3 = y->right;
     y->right = z;
     z->left = T3;
     z->height = max(getHeight(z->left),
     getHeight(z->right)) + 1;
     y->height = max(getHeight(y->left),
     getHeight(y->right)) + 1;
     return y;
```

Função para realizar uma rotação simples à esquerda

```
struct Node* rotateRight(struct Node* z)
    struct Node* y = z->right;
    struct Node* T2 = y->left;
     y->left = z;
     z->right = T2;
     z->height = max(getHeight(z->left),
     getHeight(z->right)) + 1;
     y->height = max(getHeight(y->left),
     getHeight(y->right)) + 1;
     return y;
```

Chamada de função para rotação simples

```
// Caso de rotação à esquerda direita
if (balance > 1 && compareResult < 0)
     return rotateRight(node);
// Caso de rotação à direita esquerda
 if (balance < -1 && compareResult > 0)
     return rotateLeft(node);
```

Chamada de funções para rotação a direita e esquerda

```
// Caso de rotação à esquerda direita
if (balance > 1 && compareResult > 0) {
     node->left = rotateLeft(node->left);
     return rotateRight(node);
}
// Caso de rotação à direita esquerda
 if (balance < -1 && compareResult < 0) {
     node->right = rotateRight(node->right);
     return rotateLeft(node);
}
```

E a Rubro Negro?

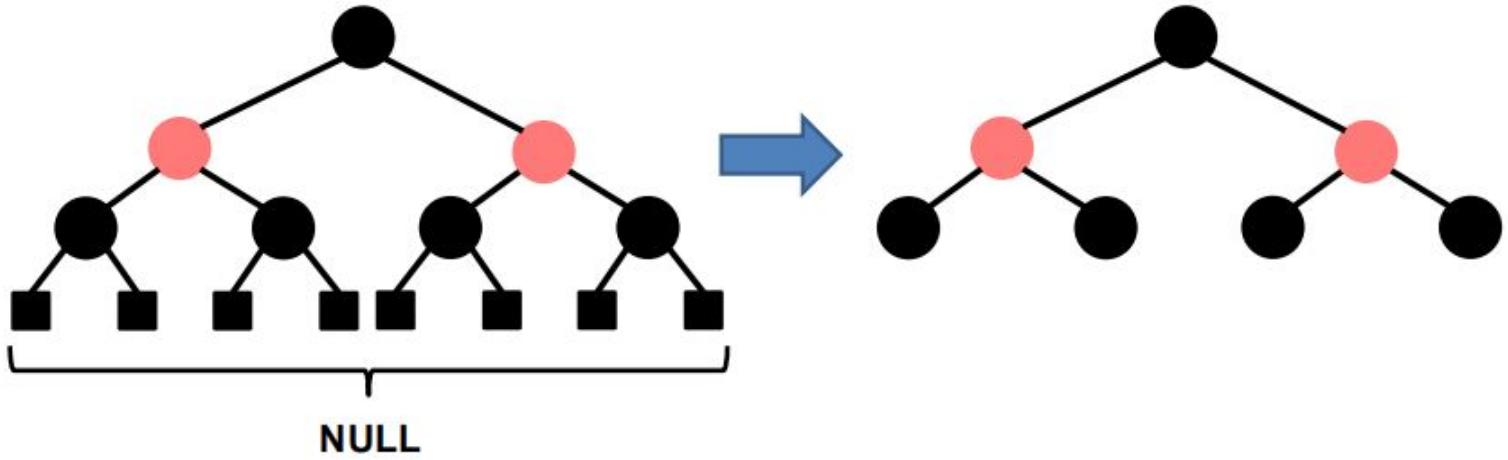

NULL

Tabela da Rubro Negra

Tabela 2: Complexidade referente a árvore Red Black

	Média	Pior Caso
Altura	*O(log n)*	*O(log n)*
Busca	*O(log n)*	*O(log n)*
Inserção	*O(log n)*	*O(log n)*
Exclusão	*O(log n)*	*O(log n)*

Aplicações

- Dicionários
- Banco de dados
- Compiladores
- Geometría Computacional
- Sistemas de Arquivo
- Rede de Computadores

Contato

arthur.teixeira@discente.ufg.br
trinde_silva@discente.ufg.br
felipe2@discente.ufg.br
kevinbrunno@discente.ufg.br
rubens.miranda@discente.ufg.br

INSTITUTO DE INFORMÁTICA UFG

UFG UNIVERSIDADE FEDERAL DE GOIÁS