Árvore Rubro-Negra   
T3 ALEST I

Enzo Russo e Raphael Remião

PUCRS

04/12/19

Porto Alegre/RS

# 1. Introdução

Este trabalho tem como objetivo descrever a árvore Rubro-Negra assim como seu funcionamento. Também serve como estudo sobre o conteúdo que foi citado em aula.

Para implementação deste trabalho, foi utilizado não somente os conhecimentos adquiridos sobre árvores, mas também sobre listas e filas, pois para alguns métodos foram necessários o uso de outros tipos de estruturas de dados.

# 2. Árvore Rubro-Negra

É um tipo de ABP (árvore binária de pesquisa) balanceada que utiliza um esquema de coloração dos nós para manter o balanceamento. Permite rebalanceamento local, ou seja, apenas a parte afetada pela inserção ou remoção de elementos da árvore é rebalanceada.

Para visualizar e percorrer uma árvore rubro-negra, não é necessário alterar nenhuma operação daquelas usadas em qualquer outra árvore binária de pesquisa. Porém, como a inserção e remoção podem violar suas **propriedades**, é preciso restaurá-la a medida que são feitas novas operações. Como, por exemplo, todo nó inserido é vermelho, utiliza-se rotações, para esquerda e para direita, e ajuste de cores para corrigir as **propriedades** que foram violadas nessa etapa.

(Como implementamos somente o método de inserção na árvore, vamos falar sobre ela em específico neste relatório).

Cada vez que chamamos essa função, testa-se a condição das propriedades e caso necessário, são efetuadas rotações e ajuste de cores até que a árvore satisfaça todo conjunto de regras. Essa rotação serve para manter o balanceamento da árvore.

Propriedades:

1. Todo nó da árvore é vermelho ou preto;
2. A **raiz** é sempre preta;
3. Toda **folha** (nil) é preta;
4. Se um nó é vermelho, então seus filhos são pretos (ou seja, não existem nós vermelhos consecutivos);
5. Para cada nó, todos os caminhos desse nó até seus nós folhas descendentes possuem o mesmo número de nós pretos.

# 3. Aplicação

Usamos como base uma árvore rubro-negra já implementada. Buscamos a mesma através do 'github' e encontramos uma que se enquadrasse nos nossos parâmetros (<https://github.com/geeksilva97/implementaca-arvore-rubronegra>).

Além dos métodos da árvore rubro-negra que utilizamos a partir da pesquisa feita, implementamos os métodos propostos pelo trabalho, aplicando as estruturas de dados previamente estudadas em aula, como fila, lista de encadeamento simples e árvore binária de pesquisa.

Para os métodos de caminhamento pós e pré-fixado e central, utilizamos lista, como implementado em aula, para retornar a ordem dos valores encontrados em cada um. Já para o caminhamento em largura, se provou necessário usarmos também um outro tipo de estrutura de dados, a fila, pois reutilizamos o método que já tínhamos implementado em aula.

Implementamos a classe CEP, que funciona como uma aplicação para testar o uso dos métodos da árvore. Optamos por uma agenda de CEPs pois nossa classe da árvore é de tipo inteiro. Criamos métodos dentro da classe da aplicação para chamar todos os métodos propostos no trabalho que foram feitos dentro da classe Arvore. Para impedirmos que o usuário colocasse qualquer número dentro da agenda de CEPs, o método geraRandom() foi concebido para que ao inserir um número inválido, esse método gera um número de CEP aleatório, porém válido.

Apesar de termos dedicado a maior parte do tempo deste trabalho em questão tentando realizar o método clone(), a complexidade de realizá-lo foi maior do que o resultado que obtivemos, pois não sabíamos como determinar qual nodo deveria ser adicionado primeiro e este foi nosso erro primário, tentar clonar a partir de inserções. Mesmo sabendo que inserindo não iria funcionar, tentar clonar diretamente foi mais difícil do que esperávamos, nos levando a não conseguir solucionar o problema. O método está implementado, porém está na forma incorreta, pois ele ele adiciona os nodos na ordem de acesso, mudando a ordem da árvore clonada em relação a original.

# 4. Conclusão

Assim, concluímos o último trabalho da disciplina, citando de forma breve e direta a árvore binária de pesquisa rubro-negra. Foi possível revistar conteúdos já trabalhados em aula além de agregarmos o objetivo proposto ao nosso conhecimento.

# 5. Referências

FEOFILOFF, Paulo. **Busca em Largura**. 22 de Agosto de 2019. Disponível em: <<https://www.ime.usp.br/~pf/algoritmos_para_grafos/aulas/bfs.html>> Acesso em: 25/11/2019

FARIAS, Lucas. **Red-Black Tree – Estruturas de Dados**. 19 de maio de 2018. Disponível em: <<https://medium.com/@luksrn/red-black-tree-estrutura-de-dados-4e13dda71280>> Acesso em: 25/11/2019.

<https://www.geeksforgeeks.org/red-black-tree-set-1-introduction-2/>

LINGUAGEM C PROGRAMAÇÃO DESCOMPLICADA. **Aula 105 – Árvore Rubro Negra – Definição**. 2016. Disponível em: <<https://www.youtube.com/watch?v=DaWNuijRRFY>> Acesso em: 02/12/2019.

AUGUSTO CÉZAR MARINHO. **Arvores Rubro negras insercao**. 2017. Disponível em: <<https://www.youtube.com/watch?v=OkGkEu5NsCs>> Acesso em: 02/12/2019.