

Prof. Carlos da Silva dos Santos

Aula prática 04 – Árvore Rubro Negra.

1 - Estude o código dos arquivos `rb_tree.h` e `rb_tree.c`.

2 - Implemente a função `left_rotate` que realiza uma rotação para esquerda na árvore, de acordo com a declaração:

```
void left_rotate(Node** T, Node* x)
```

Use a Figura 1 como referência para implementação da função. A função supõe que o nó de entrada `x` tenha um filho direito `y`. Ao final do processo, `x` torna-se o filho esquerdo de `y` e a subárvore esquerda de `y` torna-se a subárvore direita de `x`. Lembre-se que o ponteiro direito (ou esquerdo) do pai de `x` precisa ser acertado para apontar para `y`.

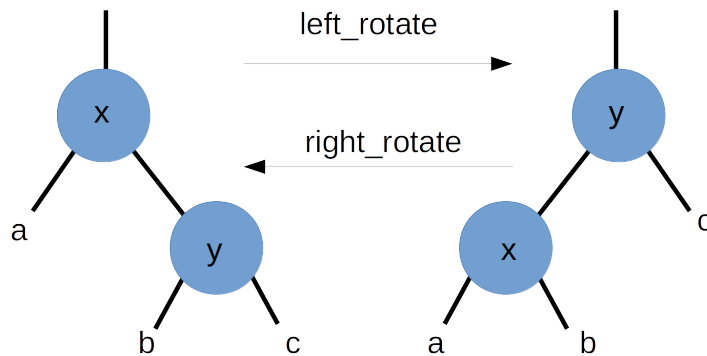


Figura 1: Ilustração das operações de rotação.

3 - Implemente a função `right_rotate` que realiza uma rotação para direita na árvore, de acordo com a declaração:

```
void right_rotate(Node** T, Node* x)
```

A função `right_rotate` é análoga à função `left_rotate` do item anterior, trocando-se esquerda por direita e vice-versa.

3 - Implemente a função `flip_color` que realiza uma coloração de nós, de acordo com a declaração:

```
void flip_color(Node** T, Node* z)
```

A troca de cores somente deve ser realizada se as seguintes condições forem verificadas:

- O nó de entrada `z` é vermelho.
- O pai (`p`) e o tio (`t`) também são vermelhos.
- O avó (`a`) de `z` é preto.

Ao final do processo, teremos:

- O nó de entrada **z** continua vermelho.
- O pai (**p**) e o tio (**t**) mudam para preto.
- O avó (**a**) de **z** torna-se vermelho.

As Figuras 2 e 3 mostram, respectivamente, uma configuração em que devemos aplicar a mudança de cor e o resultado final depois do processo.

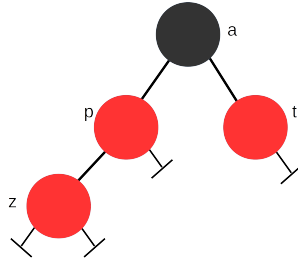


Figura 2: Configuração antes da coloração

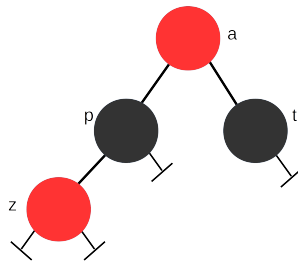


Figura 3: Configuração depois da coloração