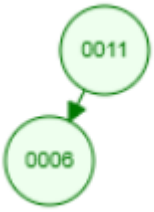
CEFET-RJ - UnED Maria da Graça

Matéria: Algoritmos e Estruturas de Dados II
Prof: Cristiano Fuschilo

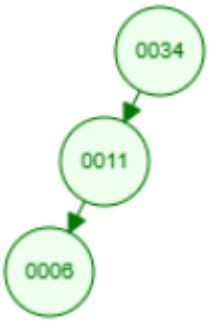
Lista de Exercícios:

1. Desenhe a árvore splay baseado nas operações abaixo:
   a. Insira o número 6;

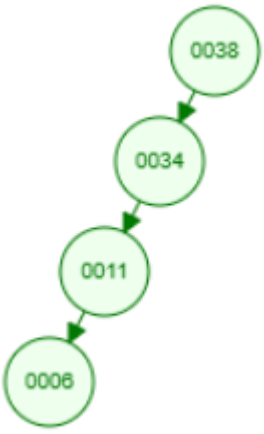
   b. Insira o número 11;



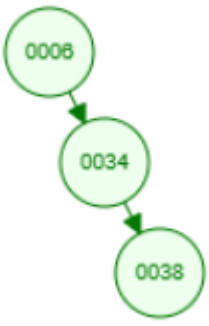
   c. Insira o número 34;



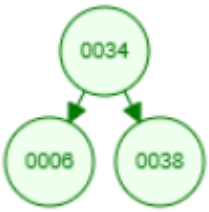
   d. Insira o número 38;



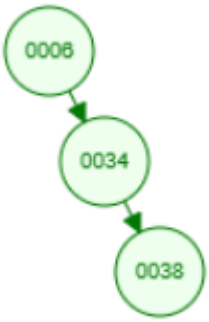
   e. Remova o número 11;



f. Busque o número 34;

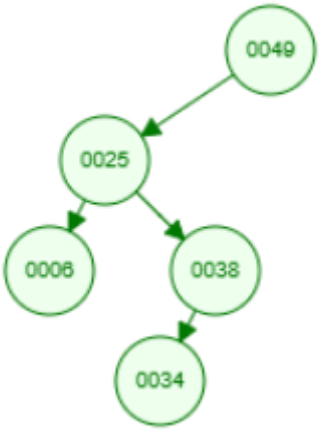


g. Busque o número 6;



h. Insira o número 25;

i. Insira o número 49;

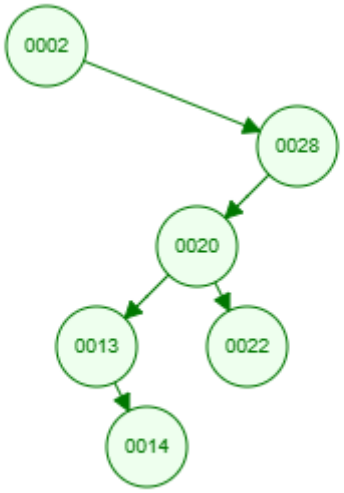


j. Remova o número 49;

2. Escreva o algoritmo de busca para uma Splay Tree.

```
No *search(struct node *root, int x)
{
    root = splay(root, x);
    if (root->value == x){
        printf("\n %d found at %p\n", root->value, root);
    }
    else{
        printf("\n %d not found\n", x);
    }
        return root;
}
```

3. Observe a árvore abaixo:



Sabendo que o n° 20 foi excluído, responda:

a. Quais serão os filhos de 28?

Left: 22

Right: NULL

b. Quem será a nova raiz?

Root: 14

c. Quem serão os filhos de 13?

Left: 2

Right: NULL

4. Implemente a função de Splay, bem como as rotações zig e zag.

```
No *splay(No *root, int x)
{
    if (root == NULL || root->value == x)
        return root;

    if (root->value > x)
    {
        if (root->left == NULL) return root;

        // Zig-Zig (Left Left)
        if (root->left->value > x)
        {
            root->left->left = splay(root->left->left, x);

            root = zig(root);
        }
        else if (root->left->value < x) // Zig-Zag (Left Right)
        {
            root->left->right = splay(root->left->right, x);

            if (root->left->right != NULL)
                root->left = zag(root->left);
        }

        return (root->left == NULL)? root: zig(root);
    }
    else
    {
        if (root->right == NULL) return root;

        // Zig-Zag (Right Left)
        if (root->right->value > x)
        {
            root->right->left = splay(root->right->left, x);

            if (root->right->left != NULL)
                root->right = zig(root->right);
        }
        else if (root->right->value < x)// Zag-Zag (Right Right)
        {
            root->right->right = splay(root->right->right, x);
            root = zag(root);
        }
        return (root->right == NULL)? root: zag(root);
    }
}
```

```c
// Rotação direita
No *zig(No *x)
{
    No *y = x->left;
    x->left = y->right;
    y->right = x;
    return y;
}

// Rotação esquerda
No *zag(No *x)
{
    No *y = x->right;
    x->right = y->left;
    y->left = x;
    return y;
}
```