

**Universidade Federal Fluminense**  
**Instituto de Computação**  
**Curso de Sistemas de Informação**  
**Estruturas de Dados para SI**

## Trabalho de implementação

Implemente uma árvore Rubro-negra capaz de armazenar em seus nós uma chave identificadora e um campo info que armazena um ponteiro genérico para um tipo que descreve alguma informação.

```
typedef struct redblack_node {int key;
                             int red; // if red=0 then the node is black
                             void * info;
                             struct redblack_node * left;
                             struct redblack_node * right;} redblack_node_t;
```

A implementação deve fornecer funções que implementam a seguinte interface:

### Inserção de um nó

```
redblack_node_t * insert(redblack_node_t * root, int key, int (*compare)(void * elem0, void *elem1));
```

### Remoção de um nó

```
redblack_node_t * delete(redblack_node_t * root, int key, int (*compare)(void * elem0, void *elem1));
```

### Busca de um nó

```
redblack_node_t *search(redblack_node_t *root,int key, int (*compare)(void * elem0, void *elem1));
```

### Destruir árvore

```
redblack_node_t *destroyTree(redblack_node_t *root);
```

### Impressão da árvore

```
redblack_node_t *printTree(redblack_node_t *root, void (*print)(void * elem));
```

As função `int (*compare)(void * elem0, void *elem1)` e `void (*print)(void * elem)` é uma callback que implementa a comparação de dois elementos genéricos apontados por `elem0` e `elem1`. A função `void (*print)(void * elem)` é uma callback que implementa como imprimir o conteúdo de um elemento apontado por `elem`.