

CI057 - Algoritmos e Estruturas de Dados III

EXERCICIO - INSERÇÃO EM ÁRVORE RUBRO-NEGRA

Considere as seguintes definições e funções para inserção de elementos em uma árvore rubro-negra.

```

enum tipoCor {RED, BLACK};

typedef struct no{
    Apontador esq, dir;
    Apontador pai;
    longInt chave;
    tipoCor cor;
} No;

RN-insere(raiz, k){
    novoNodo = criaNodo( k )
    x = raiz;
    paiX = nodoNulo;
    enquanto x <> nodoNulo
        paiX = x;
        se k < chave(x)
            x = esq(x)
        senao x = dir(x)

    pai(novoNodo) = paiX
    se paiX = nodoNulo
        raiz = novoNodo
    senao se k < chave(paiX)
        esq(paiX) = novoNodo
    senao
        dir(paiX) = novoNodo
    cor(novoNodo) = RED
    arrumaArvRN( raiz, novoNodo )
}

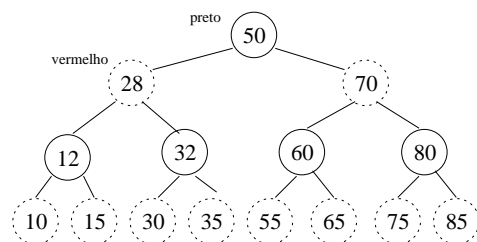
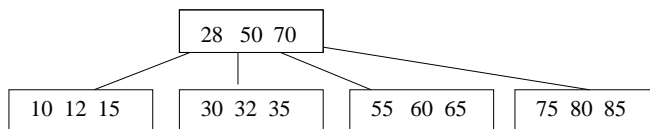
arrumaArvRN( raiz, p ){
    enquanto p != raiz E cor(pai(p)) = RED
        se pai(p) == esq(pai(pai(p))) { /* insercao na subarv.esq */
            tio = dir(pai(pai(p)))
            se cor(tio) == RED {
                cor(pai(p)) = BLACK /* Caso 1 */
                cor(tio) = BLACK
                cor(pai(pai(p))) = RED
                p = pai(pai(p))
            }
            senao {
                se p == dir(pai(p)) {
                    p = pai(p) /* Caso 2: esq-dir */
                    rotEsq(p)

                    cor(pai(p)) = BLACK /* Caso 3 esq-esq*/
                    cor(pai(pai(p))) = RED
                    rotDir( pai(pai(p)) )
                }
                senao { /* insercao na subarv.dir */
                    tio = esq(pai(pai(p)))
                    se cor(tio) == RED {
                        /
                        cor(pai(p)) = BLACK /* Caso 1 */
                        cor(tio) = BLACK
                        cor(pai(pai(p))) = RED
                        p = pai(pai(p))
                    }
                    senao {
                        se p == esq(pai(p)) {
                            p = pai(p) /* Caso 2: dir-esq */
                            rotDir(p)

                            cor(pai(p)) = BLACK /* Caso 3 dir-dir*/
                            cor(pai(pai(p))) = RED
                            rotEsq( pai(pai(p)) )
                        }
                    }
                }
            }
        }
    cor(raiz) = BLACK
}

```

Simule a inserção dos elementos abaixo nas árvores 2-3-4 e rubro-negra e acompanhe o trecho do código.
 Caso 1: "split" de nodo-4: insercao de 5

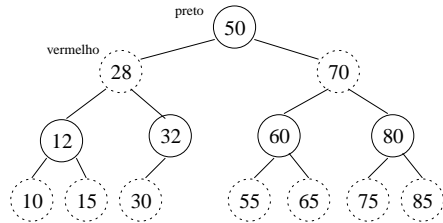
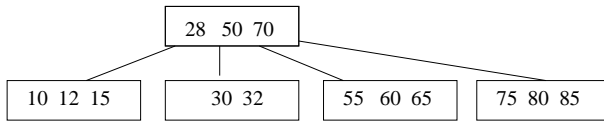


```

enquanto p!=raiz E cor(pai(p)) = RED
    se pai(p) == esq(pai(pai(p))) {
        tio = dir(pai(pai(p)))
        se cor(tio) == RED {
            cor(pai(p)) = BLACK
            cor(tio) = BLACK
            cor(pai(pai(p))) = RED
            p = pai(pai(p))
        }
        ... }
    cor(raiz) = BLACK

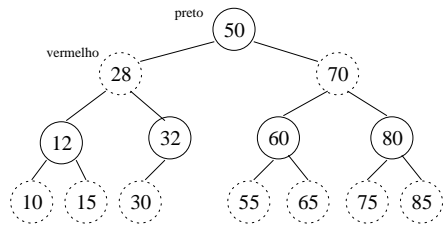
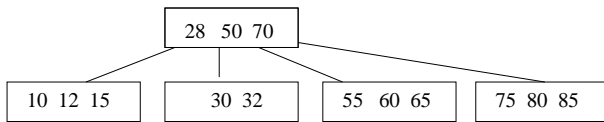
```

Caso 3: insercao de elemento em nodo tipo-3 : insercao de 29



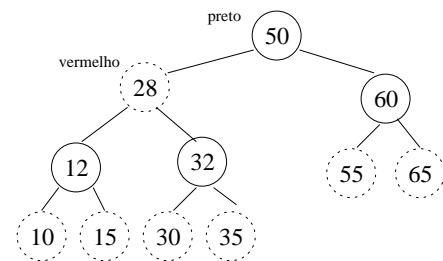
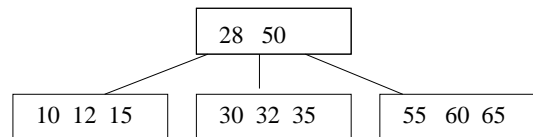
```
cor(pai(p)) = BLACK /* Caso 3 esq-esq*/
cor(pai(pai(p))) = RED
rotDir( pai(pai(p)) )
```

Caso 2: insercao de elemento em nodo tipo-3 : insercao de 31



```
se p == dir(pai(p)){
  p = pai(p) /* Caso 2: esq-dir */
  rotEsq(p)
}
cor(pai(p)) = BLACK /* Caso 3 esq-esq*/
cor(pai(pai(p))) = RED
rotDir( pai(pai(p)) )
```

Caso 1 e 2: "split" de nodo tipo-4 :e insercao em nodo tipo-3: insercao de 33



Responda:

1. Considerando que a quantidade de elementos armazenados na árvore é n , qual a altura máxima de uma árvore rubro-negra?
2. Quantas vezes no maximo o Caso 1 é executado?