

```

// Inserimento di un nodo in un albero Red-Black
insertNode(TREE T, TREE k, ITEM x)
    TREE p ← nil // riferimento al padre
    TREE u ← T // riferimento alla radice

    // cerco posizione inserimento
    finché u ≠ nil and u.key ≠ k fai
    |
    |   p ← u
    |   u ← iif(k < u.key, u.left, u.right)
    |
    se u ≠ nil and u.key == k allora
    |   // la chiave è già presente, aggiornò il valore
    |   u.value ← v
    altrimenti
    |   // la chiave non è presente
    |   // creo un nodo coppia chiave-valore
    |   TREE new ← Tree(k, v)
    |
    |   // collego il nodo creato
    |   link(p, new, k)
    |   balanceInsert(new)
    |
    |   se p == nil allora
    |   |   T ← new // primo nodo ad essere inserito
    |
    // restituisco l'albero non modificato o il nuovo nodo
    ritorna T

```