

```
// effettua una rotazione verso destra
rotateRight(TREE p, TREE u, ITEM x)
    // entrambi potrebbero essere nil
    (1) TREE y ← x.left
        TREE p ← x.parent
    (2) x.left ← y.right // il sottoalbero B diventa figlio sinistro di x
        se y.right ≠ nil allora
            y.right.parent ← x
    (3) y.right ← x // x diventa figlio destro di y
        x.parent ← y
    (4) y.parent ← p // y diventa figlio di p
        se p ≠ nil allora
            se p.right == x allora
                p.right ← y
            p.left ← y
ritorna y
```