

```
// effettua una rotazione verso destra
TREE rotateRight(TREE x)
    // entrambi potrebbero essere nil
    TREE y ← x.left
    TREE p ← x.parent
    (1) x.left ← y.right // il sottoalbero B diventa figlio sinistro di x
    if y.right ≠ nil then
        (2)   y.right.parent ← x
    y.right ← x // x diventa figlio destro di y
    x.parent ← y
    (3) y.parent ← p // y diventa figlio di p
    if p ≠ nil then
        (4)       if p.right == x then
            (5)           p.right ← y
        p.left ← y
    return y
```