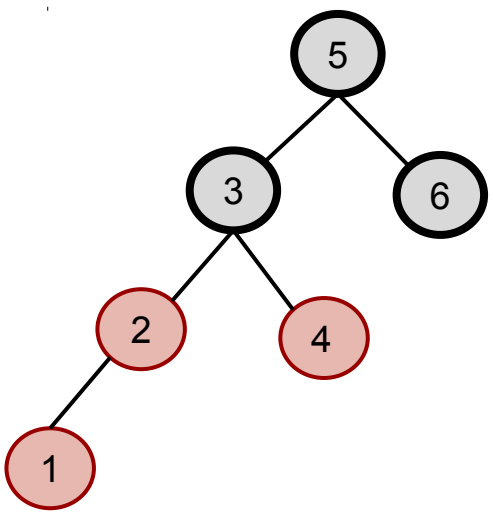
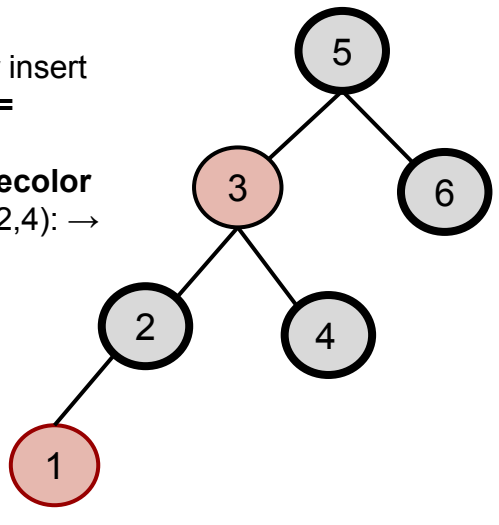
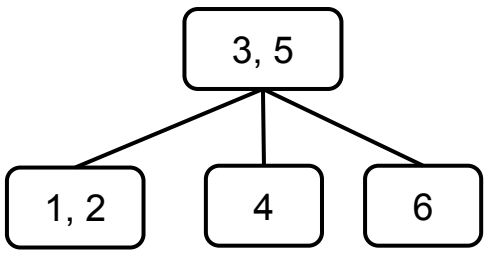
Solutions: Balanced Trees

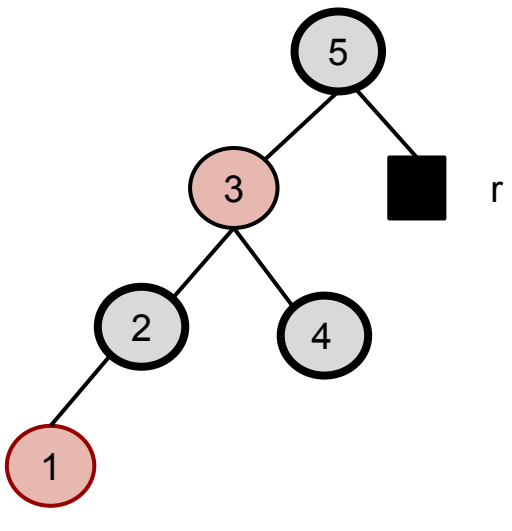


Insert 1:
← (a) RBT after insert
(b) **double red = overflow**
(c) **red uncle: recolor**
(d,e) RBT and (2,4): →

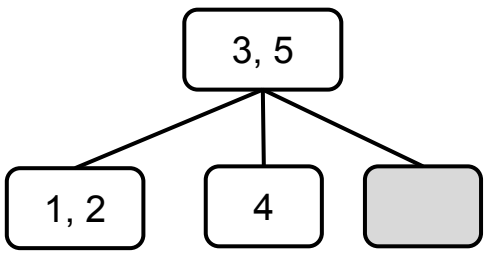




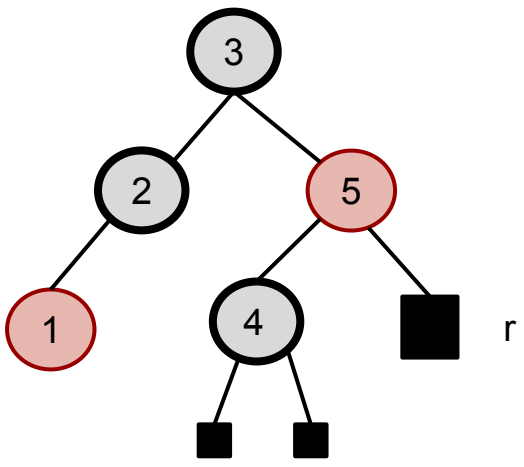
DELETION



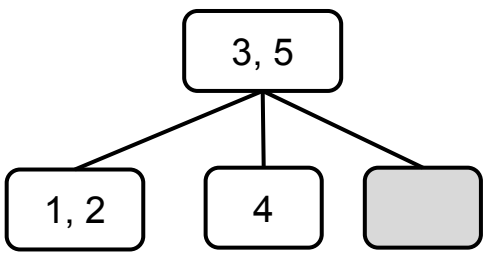
Delete 6:
(b): **double black at r**, corresponding to **underflow**
(c): **<u>Case 3</u>: sibling "3" is red: perform RBT <u>adjustment</u>.**
(Another case will then apply.)

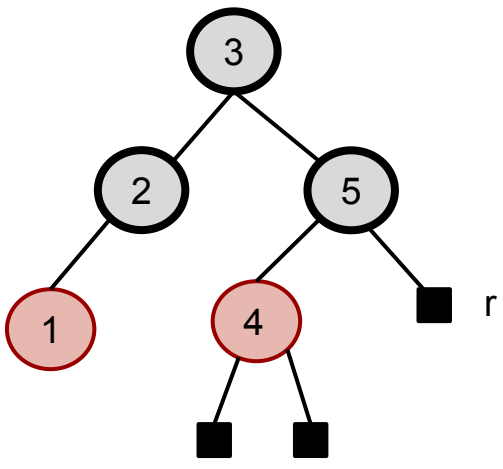


(Note: the text below departs from the template given students.)



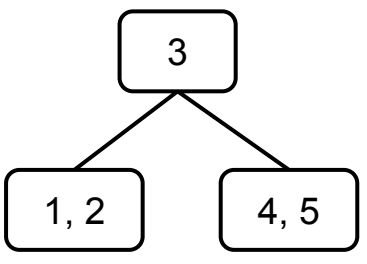
New Case:
(b): still **double black at r**, corresponding to **underflow,** but now ...
(c): **<u>Case 2</u>: sibling "4" is black with black children: perform RBT <u>recoloring</u>, equivalent to 2-4 fusion.**
(Essentially, push the blackness of both 4 and r up to parent 5.)

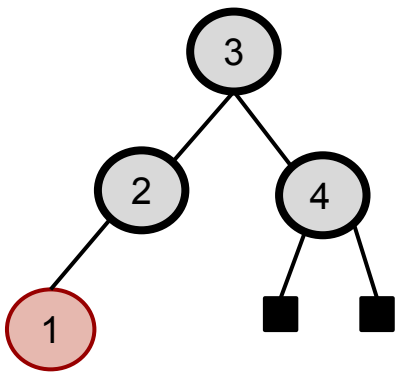




Now we are good. A red node absorbed the extra black by turning black.

(if 5 had been black the double black violation would propagate up the tree, potentially O(lg n) times.)





Delete 5:
BST deletion says when there is one child, replace it with its child.
RBT deletion says the child should take on its parent's color.
All is good.