* Describe the right rotate routine used in Red-Black trees.  
  + Set X to root.
  + Replace B with Y (Y becomes X’s right child)
  + B becomes Y’s left child.
* What are the properties of Red-Black trees.
  + A type of binary search tree (used to ensure balancing in a tree)
  + Searches in O(log n) time in worst case
  + Every node is red or black
  + Root is always black
  + Every leaf node is black
  + If a node is red, both its children are black
  + For each node, all paths from the node to descendant leaves contain the same number of black nodes.
* Suppose while making a purchase you wish to use the minimum number of coins.  If you have plenty of quarters dimes, nickles, and pennies, what would a greedy approach to solve this be?  Is it optimal?
  + The most straightforward greedy approach for this problem would be to select the largest denomination of coin available that is less than the remaining amount consecutively until the goal amount is reached. This is optimal for some currencies but not all. For American change, it is optimal.   
    To play devil’s advocate, imagine a currency system where your coin denominations are 1, 3, and 4. If you wanted to make 6, this algorithm would choose (4,1,1) where in actuality the optimal solution is (3,3).
* Describe how the dynamic programming rod cutting algorithm works.
  + The dynamic approach is proposed because naïve recursive solutions are highly inefficient due to the fact that they solve the same subproblems over and over. In a dynamic programming approach, we can solve them only once, and refer back to the subproblem’s solution if needed later. The approach trades computation time for memory.
  + TOP DOWN APPROACH:
    - Write the rod cutting procedure recursively but modified to save the results of each subproblem. The procedure should first check if it has already solved the subproblem. If so, return the saved value. If not, compute the value normally. This is a *memorized* approach.
  + BOTTOM UP APPROACH
    - Sort all subproblems by size and solve in smallest size order first. As you solve subproblems, you have already solved all smaller subproblems the solution depends on. Each subproblem is only solved once, and when seen all prerequisite subproblems have been solved already.
* In a B-tree, what are the two ways the keys in internal nodes used?