Homework #3

P1: (a) Draw the resulting binary search tree after inserting [64, 21, 35, 634, 29, 42, 13, 53, 32].

(b) What is the average node depth of this tree?

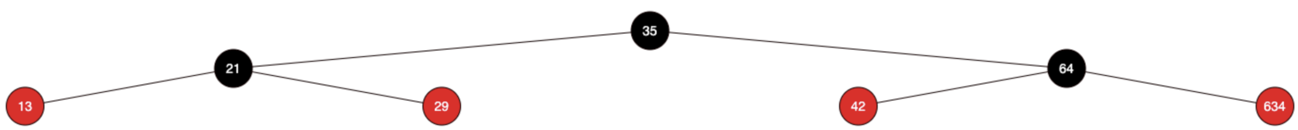
(c) Draw a balanced binary search tree with these numbers.

(d) Delete 21 and then 35 from the (a). Draw the resulting tree.

P2: (a) Draw the resulting *red-black* tree after inserting [64, 21, 35].

(b) Continue inserting [634], draw the resulting RB tree.

(c) Continue inserting [29, 42, 13], and you will get the following.



Now insert [53, 32] to it. Draw the resulting RB tree.

P3: In the class, we discussed about the `successor` function for binary search tree.

Write the `predecessor` procedure in pseudocode.

P4: Show that if a node in a binary search tree has two children, then its successor has no left child and its predecessor has no right child.

P5: What is the largest possible number of internal nodes in a red-black tree with black-height k? What is the smallest possible number? The black-height is the number of black nodes in a simple path excluding the NIL leaves and including the root node.

P6: Consider a red-black tree formed by inserting n nodes with `RB-INSERT`. Argue that if $n > 1$, the tree has at least one red node.

P7: Consider a binary search tree allowing duplicate numbers. Let's call it $DBST$. Assume it can defined as follows.

1. The left subtree of a node contains only nodes with keys less or equal to the node's key
2. The right subtree of a node contains only nodes with keys greater or equal to the node's key
3. The left and right subtrees must also be $DBST$.

In other words, the keys of one node and its children can be the same. Write a procedure in pseudocode that can find the *mode*, ie. the value that appears most, in a DBST. If there is more than one modes, you can report any one of them.