Informatics II, Spring 2023, Exercise 9

Publication of exercise: May 01, 2023
Publication of solution: May 7, 2023
Exercise classes: May 8 - 12, 2023

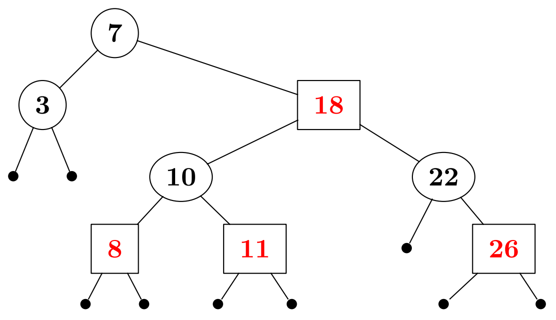
Red Black Tree

Consider the red-black tree in lecture where black nodes are denoted with a circle and red nodes are denoted with a square. We use the key to represent a node. For example, 2 represents the node with key 2. We consider four operations:

1. Creating a new node (left and right are set to NULL):
   Example: create node 9: `create(9)`
2. Setting a property (color, key, left, right) of a node:
   Example: set the key of node 9 to 5: `9->key = 5`. Here, 9 represents the node with key 9.
3. Rotating a node:
   Example: right rotate node 5: `RightRotate(5)`
4. Deleting a node:
   Example: delete node 9: `delete(9)`

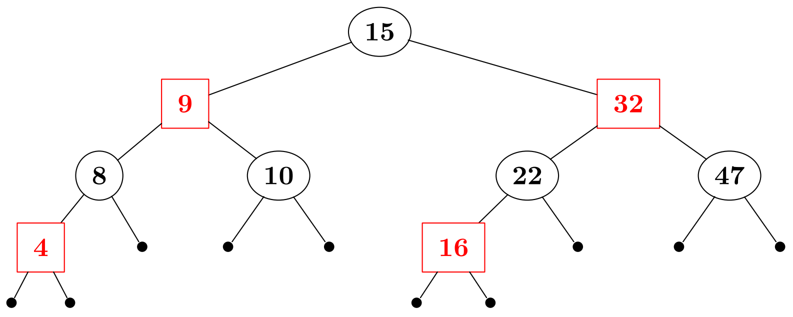
Task 1. Consider the red-black tree shown in following figure. State the operations and stepwise changes in tree when 15 is inserted in the tree (You have to show the state of tree after color change and rotation transformations).

Initial tree



Task 2. Consider the red-black tree shown in Figure below. Perform the following operations on it and draw the resulting tree after each of them: Delete 10, Insert 5, Delete 9, Delete 15.

Initial tree



Task 3. You are given an implementation of binary search tree in the file `task3_framework.c`. A **binary search tree** is of the following type:

```
struct TreeNode {
  int val;
  struct TreeNode* left;
  struct TreeNode* right;
  struct TreeNode* parent;
};
```

The implementation also includes the following functions:

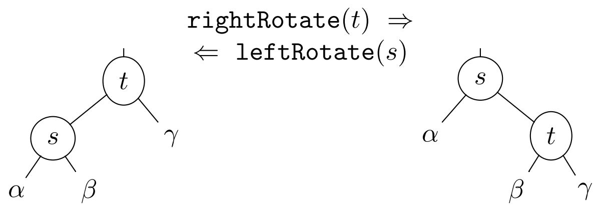
- *struct TreeNode* insert(struct TreeNode* root, struct TreeNode* parent, int val)* - inserts an integer `val` into the binary search tree.
- *struct TreeNode* search(struct TreeNode* root, int val)* - search and returns a binary search tree node with value *val*. Return *NULL* if value is not found.

Your task is to implement two functions in C:

Task 3.1 *struct TreeNode* leftRotate(struct TreeNode* root, int val)* - that left rotates the binary search tree node with value *val*.

Task 3.2 *struct TreeNode* rightRotate(struct TreeNode* root, int val)* - that right rotates the binary search tree node with value *val*.

The left and right rotation operations on nodes s and t respectively have been illustrated in Figure below where α, β and γ represents the subtrees.


rightRotate(t) ⇒
⇐ leftRotate(s)
t
s
γ
α
β
s
α
t
β
γ