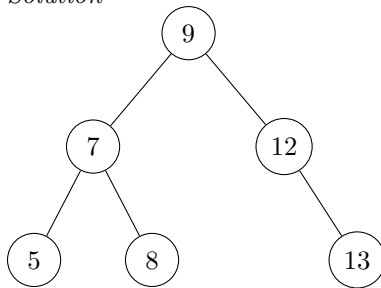


Assignment 03**Question 1**

Draw the resulting AVL tree from the following commands

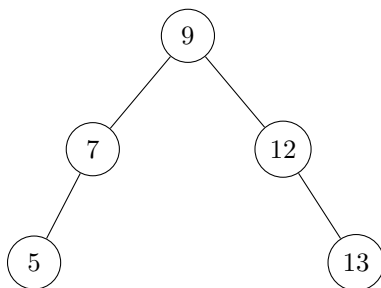
Insert: 5, 7, 8, 9, 12, 13

Solution



Delete: 8

Solution

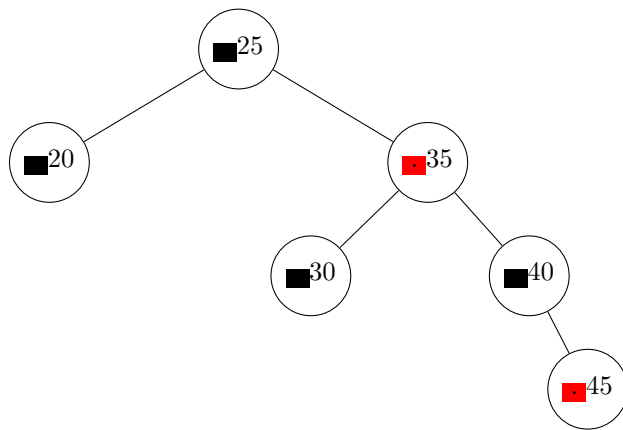


Question 2

Provide an example of a legal red-black tree configuration that is NOT a legal AVL tree configuration

Solution

Insert: 20, 25, 30, 35, 40, 45



Question 3

Write a pseudocode for deleting the node with the minimum value in an AVL tree

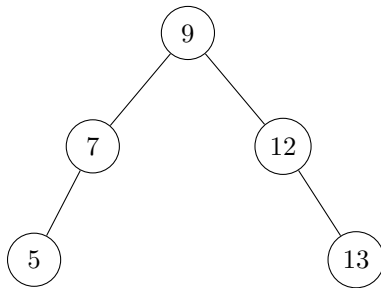
Solution

```
function MIN(Node node)
  if node.left == null then
    return null
  else if node.left != null then
    return min(node.left)
  end if
  Balance()
end function
```

Question 4

Refer to the `binarytree.pdf` file that was provided on the Google Drive. Draw the array representation of the resulting tree you drew in Q1. From within the array itself, how could you find the parent, rightchild and leftchild of that node by only working with the array representation?

Solution



Array representation: (7, 5, 12, null, null, 9, 13)
our array starts from index = 0 to index = 6

From the `binarytree.pdf` file

if a node's index is i
 \therefore parent's index = $\frac{i-1}{2}$
 \therefore left child's index = $2i + 1$
 \therefore right child's index = $2i + 2$

PROGRAMMING PART

Question 5

Threading Part 1

Write a program that makes use of threading. You will have one thread count how many times a second has passed and the second thread will count how many times 3 seconds has passed. This will occur in an infinite loop and the counts will be printed to the screen. Indent the count of one of the threads for easier viewing.

Question 6

Threading Part 2

Write a program that has four threads. These four threads will share use of an array. Let each thread take turns adding a random number to the array. Each thread should wait its turn and not be doing an insert when another thread is currently doing the insert. Print to the screen each action of each thread.

Question 7

Working with files Part 3

Write a program that reads in two different files and compares each line of the file. If the lines are different, output that they are different to an output file called “diff.txt”. Do not print any output to the screen, only to the output file.