Student: John Rizzo

Course: CS590-A Algorithms

Instructor: Dr. William Hendrix

Due Date: February 27, 2025

Description: Homework 3 Algorithms

Problem 1

What new field(s) does the data structure need?

The new solution requires that the root node is augmented to store the minimum value, such as in node.minval.

Problem 2

Give pseudocode for the min operation for the BST.

Algorithm 1 BST.min()

Output: The minimum value in the tree

```
node = root
if node ≠ NIL then
    return node.minval
end if
```

Problem 3

Give pseudocode for the insert operation. Reference pseudocode for the insert method appears below.

Algorithm 2 BST.insert()

```
node = root
while node ≠ NIL do
    if node.value ≤ new then
        if node.left = NIL then
            Add new as left child of node
            node = node.left
        else
            node = node.left
        end if
        if root.minval > node.value then
            root.minval = node.value
        end if
    else
        if node.right = NIL then
            Add new as right child of node
            node = node.right
        else
            node = node.right
        end if
    end if
end while
```

Problem 4

Algorithm 3 BST.delete(node)

```
if node has two children then
    swapnode = right
    while swapnode has a left child do
        swapnode = swapnode.left
    end while
    Swap node's parent and children links with swapnode
    if node is the BST root then
        Set root to be swapnode
    end if
end if
if node has no children then
    if node is the root then
        Set root to be NIL
    else
        Set node.parent's child to be NIL
    end if
else
    // node must have one child
    if node is the root then
        Set root to be node's child
    else
        Set node.parent's child to be node's child
    end if
    Set node's child's parent to be node.parent
    Find the minimum value from the root
    Set the root's min to be the minimum value
end if
```

Problem 5

Give pseudocode for an eficient algorithm for the *top-k* search problem. In top-k search, you are given an array of n integers and must return the k largest integers, where k is generally much smaller than n. Acceptable algorithms might be $O(n + klgn)$ or $O(nlgk)$, but not $O(nk)$ or $O(nlgn)$. *Hint* use an appropriate data structure!

Algorithm 4 top-k Search

$heap = \emptyset$
$result = \emptyset$
for $i = 0$ to n **do**
    Insert $arr[i]$ into $heap$
end for
for $i = 1$ to $k + 1$ **do**
    $max = heap.max()$
    $heap.delete(max)$
    $result.insert(max)$
end for
return $result$