

Homework 10

Due 10/05/16

September 30, 2016

Describe a modification of a Binary Search Tree dictionary that can return the minimum value in the dictionary in $\Theta(1)$ time. This change should *not* increase the asymptotic complexity of any other dictionary operation.

1. What new field(s) does the data structure need?
2. How does this change impact the min, insert, and delete methods of the BST? Note that insertion and deletion may change the minimum value in the BST.

Reference implementations of the min, insert, and delete functions appear below.

```
1 Algorithm: BSTDict.min()
2 node = root
3 while node has a left child do
4   |   node = node.left
5 end
6 return node
```

```

1 Algorithm: BSTDict.insert(new)
2 node = root
3 while node isn't NIL do
4   if node.value ≤ new then
5     if node.left = NIL then
6       Add new as left child of node
7       node = node.left
8     end
9     node = node.left
10  else
11    if node.right = NIL then
12      Add new as right child of node
13      node = node.right
14    end
15    node = node.right
16  end
17 end

```

```

1 Algorithm: BSTDict.delete(node)
2 if node has two children then
3   | swapnode = right
4   | while swapnode has a left child do
5   |   | swapnode = swapnode.left
6   | end
7   | Swap node's parent and children links with swapnode
8   | if node is the BST root then
9   |   | Set root to be swapnode
10  | end
11 end
12 if node has no children then
13   | if node is the root then
14   |   | Set root to be NIL
15   | else
16   |   | Set node.parent's child to be NIL
17   | end
18 else
19   | /* node must have one child */
20   | if node is the root then
21   |   | Set root to be node's child
22   | else
23   |   | Set node.parent's child to be node's child
24   | end
25   | Set node's child's parent to be node.parent
26 end

```