

1. The data structure needs a new field to store the current minimum value of the Binary Search Tree. In the program, we'll call this variable 'min'.
2. The following changes are needed to return the minimum value in the Binary Search Tree:

```

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

```

```

1 Algorithm: BSTDict.insert(new)
2 node = root
3 Boolean bool = false;
4 while node isn't NIL do
5     if node.value ≤ new then
6         if node.left = NIL then
7             Add new as left child of node
8             node = node.left
9         end
10        node = node.left
11    else
12        bool = true;
13        if node.right = NIL then
14            Add new as right child of node
15            node = node.right
16        end
17        node = node.right
18    end
19    if bool is false then
20        Set min to be new;
21    end
22 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 min then
14         Set min to be node.parent
15     if node is the root then
16         Set root to be NIL
17     else
18         Set node.parent's child to be NIL
19     end
20 else

```

```
21  /* node must have one child */
22  if node is the min then
23      Set min to be node.right
24      while node.right has a left child
25          min = node.left
26      end
27  end
28  if node is the root then
29      Set root to be node's child
30  else
31      Set node.parent's child to be node's child
32  end
33  Set node's child's parent to be node.parent
34 end
```