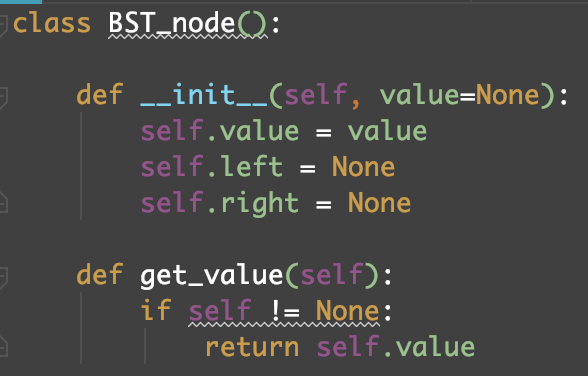
Lab 1 Report Template

1. **Given elements 1, 2, 3, 4, 5, 6, 7. Implement BST algorithms.** 
   1. Every node in the BST has two child nodes, which are called left-node and right-node. All of the nodes in the BST could point to None value.



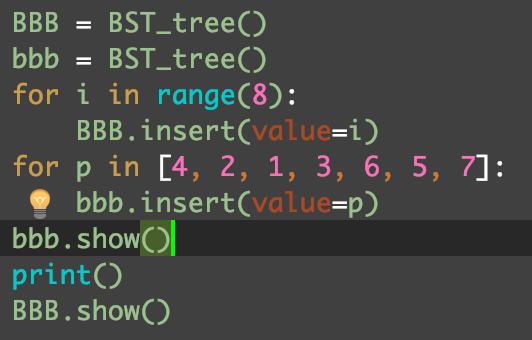
* 1. In the BST tree, only the function ***insert*** is used to add a new node into the tree. While the head node does not has value, the value will be added to the head node of the tree. In the other conditions, the value will search for a node whose child node that suitable for the value is point to a None value.



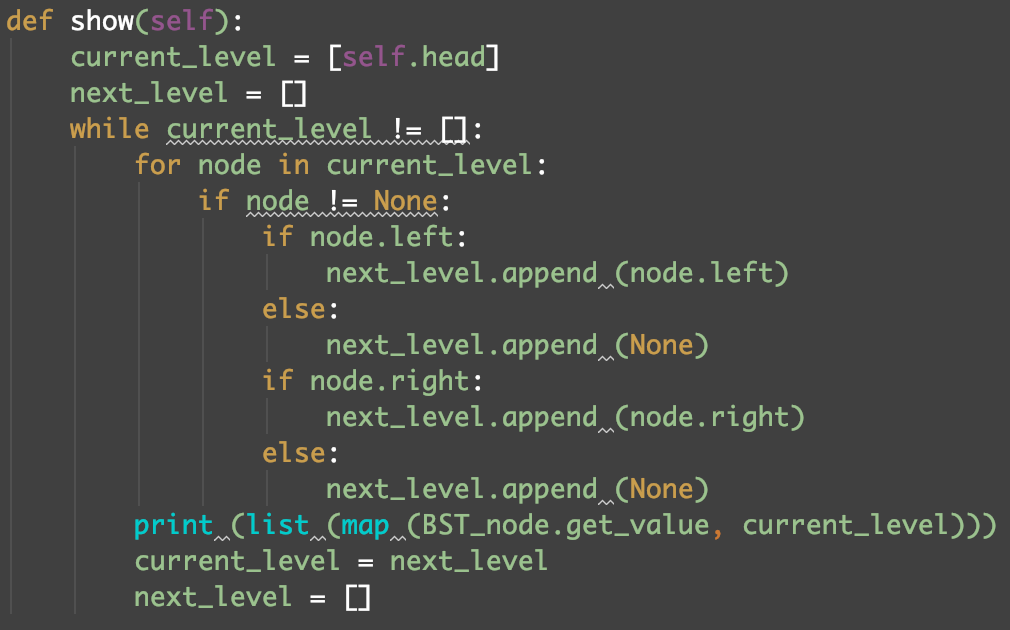
1. **Insert them into a binary search tree in the orders:**

**4, 2, 1, 3, 6, 5, 7 and 1, 2, 3, 4, 5, 6, 7**

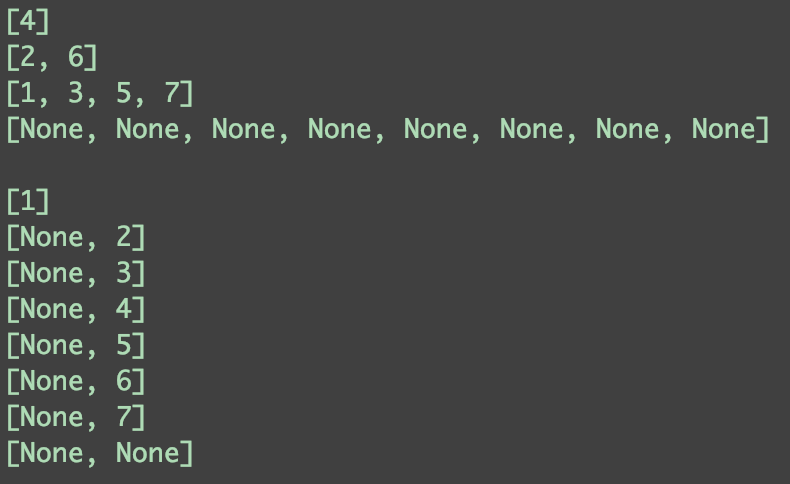
* 1. Test code:



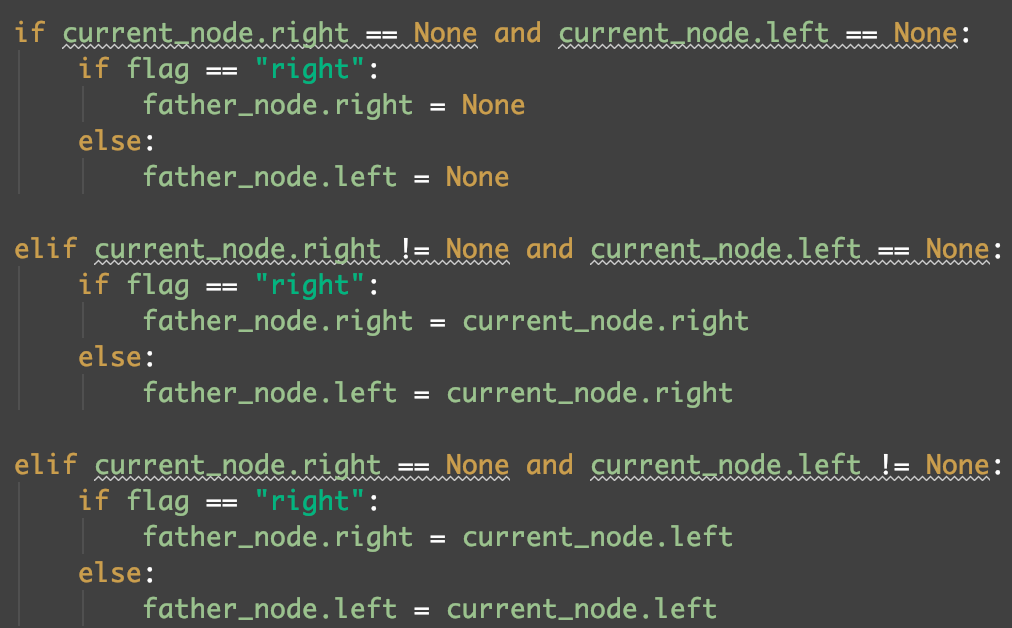
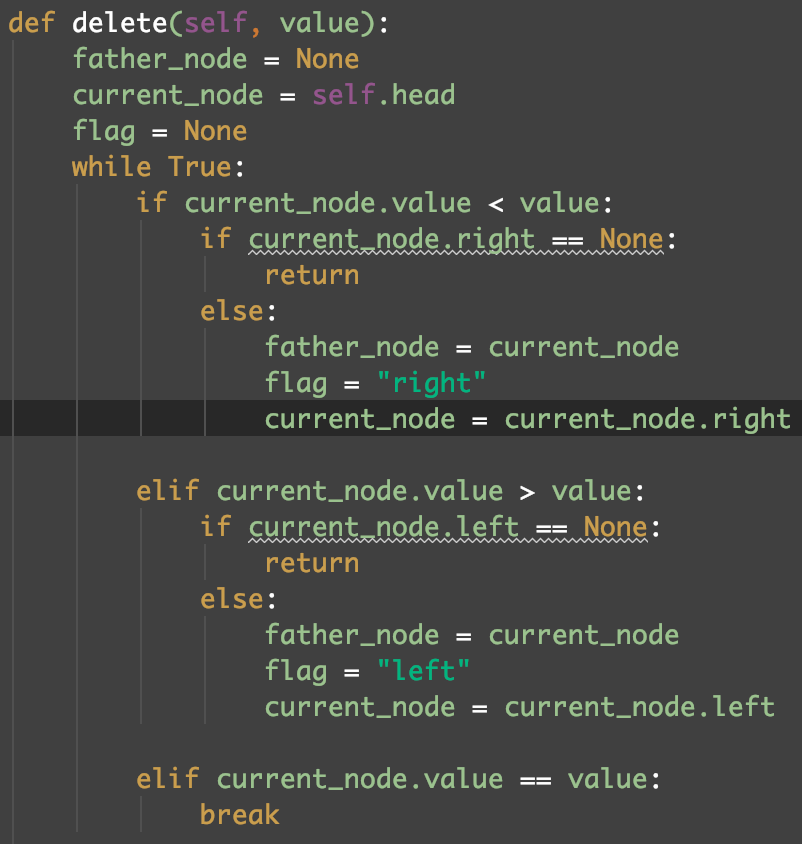
* 1. I use a method called ***show*** to display the tree.

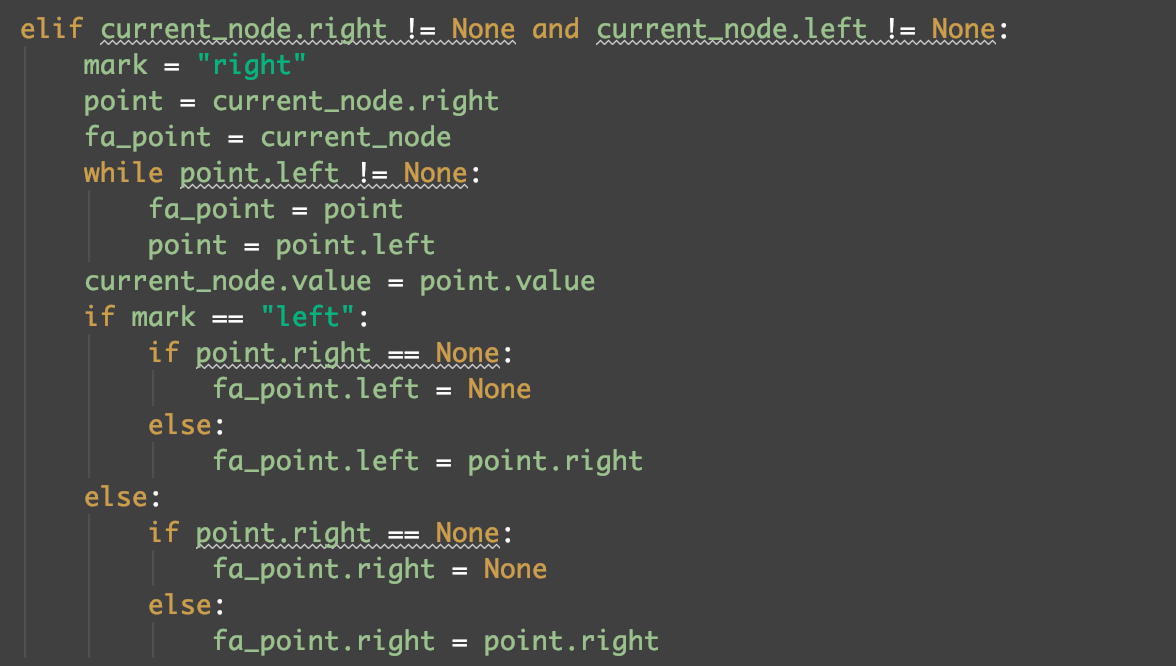


* 1. Output of the test:

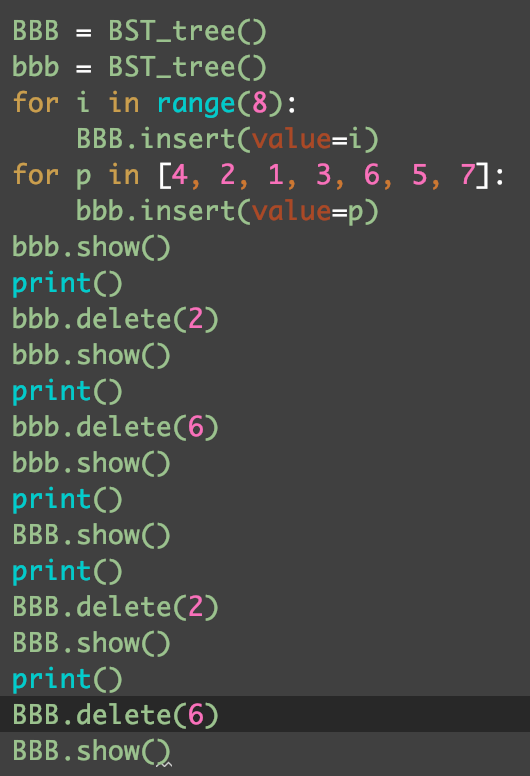


1. **Delete element 2, and then delete element 6**
   1. The ***delete*** method:

****If we want to delete a node, we also have to find the father node of it, since while the node does not have 2 child node, we need to do some changes in its father node. If there is no child, the father just point to None. Also, if there is only one child just let the father point to the child. While there are two children, just delete the node containing the minimum of the right-subtree and use the minimum to replace the value of the node.

****

(2) The Test code:



(3) Output:

