**Note**: 2 out of 4 lab assignments will be graded and counted towards 20% of the course. Python is the ONLY accepted programming language for this course.

**WARNING**: disciplinary actions (zero mark for the lab, or immediate failure of the course, or academic warning from the university) will be taken for any plagiarism.

**Due time**: Friday, 6th April, 23:59PM through NTULearn -> MH1402 -> Labs -> LAB3 Submission. You may submit multiple times, only the last version counts. Indicate your Matric Number in all your submission files.

**Task**: Implementation of Binary Search Tree with the provided TreeNode class, it should support the following functions:

- search(key): given a key value, search through the tree, and return the node with such a "key" value; return None if not found.

- insert(key): create a new node with "key" value and insert it to the tree, one should search first for the location where it should be inserted to, followed by the insertion procedure.

- delete(key): delete the node with "key" value, do no deletion if no such node is found. For replacement:
  - the node does not have any child, no replacement is needed
  - the node has only one child, push the subtree rooted by the child one level up
  - the node has two children, replace it by the leftmost node in the right subtree (note the leftmost node may have a rightChild, which needs to be pushed up by one level also)

  Note: if the root is changed in the process, the new root needs to be updated to the "root" variable of this class.

- searchByRange(minimum, maximum): return **the list of nodes** of key values in between [minimum, maximum] both inclusive. One can assume the given parameter has already enforced minimum <= maximum. This function should first locate the node with the smallest possible key value such that key value >= minimum, followed by looking for all elements by an in-order traversal until the key value of nodes are larger than maximum.