**World Quant University**

**Professor: Harry Wang**

**Algorithms II**

Nikolas Lippmann Pareschi - [nikolaslippmann@gmail.com](mailto:nikolaslippmann@gmail.com)

**Assignment 3: B-Trees**

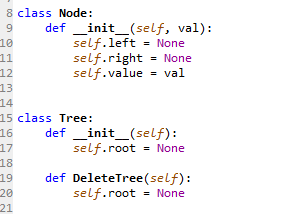
**Problem :  Make Changes to the B-Tree Code**

Given an implementation of a B-Tree code structure, continue to use Python to do the following:

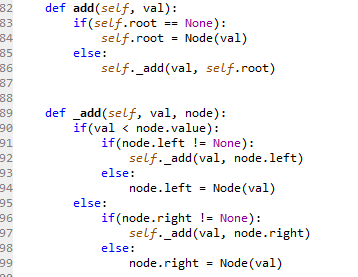
* + - * 1. Implement the function \_\_print\_tree, this function should print all keys in the B-Tree in increasing order. For example.. the output of the test example (after completing all insert operations) in the main part in the code is:  B-Tree: 0 1 2 3 4 5 6 7 8 9

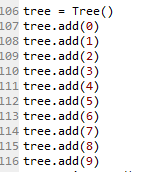
First we implement our classes:

Code:

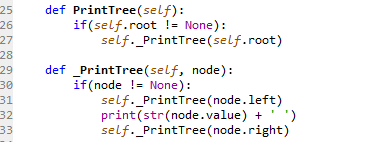


Then we add the values:

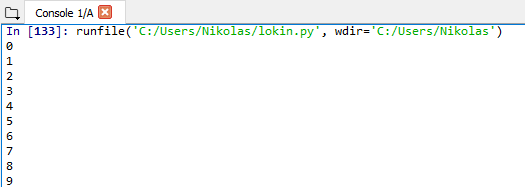




We define the print functions:

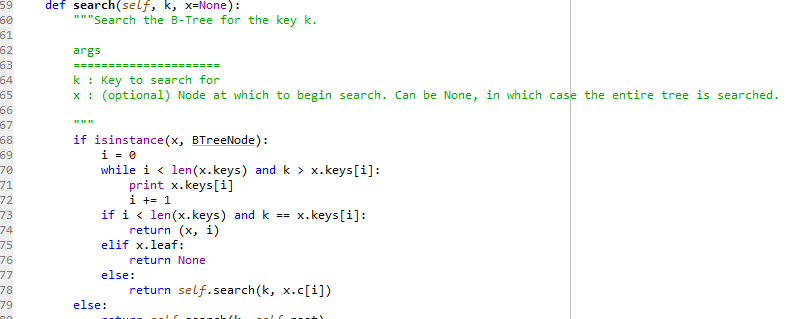


And we print the values in the tree:



* + - * 1. Implement the function  def search(x, k, nil=None) which should return a pointer to the node with key=k in the B-tree T where x is the root node in the B-tree T.
        2. Update def search(x, k, nil=None) function to print all keys which would be compared to the value k before returning a pointer to the node in B-Tree T with key=k.

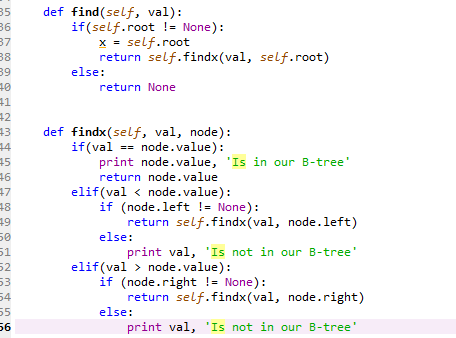
Answering question 2 and 3:



I was unable to run this in combination with my code. But I understand that this is the true objective of the question, I receive this error in the console and was unable to fix:



Probably we are entering in some infinite loop. Nevertheless the code is able to check if a key is present in the tree:







My algorithm is based in 2 other algorithms, one provided in

<https://gist.github.com/natekupp/1763661>

And the other provided in:

<http://www.damiantgordon.com/Videos/ProgrammingAndAlgorithms/Code/BinaryTrees.py>