18560 - Uma Sankar

/* Evaluation:

1. Code compilation:

1. Does code compile without errors? - Yes.
2. Was a readme.txt file included with instructions on how to compile and run? - yes.

2. Correctness (positive test cases):

1. Can I insert a key? No. Insertion is failing after inserting 2 keys:

```
c:\Krishna\sssihl\Database Internals Course\2019\Lab Assignments\01-BPlus Tree\Uma_18560>
java -cp . BPlusTree keys2.txt
Inserting Key     File Index
=============     ==========
  A               0
  C               1
Exception in thread "main" java.lang.NullPointerException
      at java.lang.String.compareTo(String.java:1155)
      at LeafNode.insertInLeaf(LeafNode.java:39)
      at BPlusTree.insert(BPlusTree.java:141)
      at BPlusTree.main(BPlusTree.java:248)
```

2. Can I delete a key? - not implemented.
3. Can I search for a key? - Unable to test since insertion is failing on launch.
4. Can I view display of tree? - No.
5. Can I specify size of B+ tree node (# of keys in a node)? - No. This is hard coded in the Node class:

```
public abstract class Node
{

        int n = 4;
        String keys[] = new String[n];
        int size;
        Node parent;

        public abstract boolean isLeaf();

}
```

6. Do the nodes satisfy the B+ tree property? - Unable to test since insertion is failing on launch.
7. Can I create a B+ tree from a file of keys? - Option exists.
8. Can I save my B+ tree to a file? - no.
9. Can I load back the file saved in step 7? - No.
10. Can I insert and delete keys from the command line even after loading keys from file? - No.
11. Is Output for keys1.txt correct? - No
12. Is output for keys2.txt correct? - No

3. Programming Style & General Comments:

1. Are there useful comments that complement the code? - No
2. Is the indentation style neat and consistent? - Yes.

3. Are there had coded limits or magic numbers used in the code? - Yes. Size of node is hard coded to 4.
4. Are there hard coded file paths used in the code?

4. Exception Handling:

1. Delete on empty tree - not implemented
2. Delete a non-existent key - not implemented.
3. Insert a key that exists already (keys3.txt) - fails with java.lang.NullPointerException.
4. Call display on empty tree - display not implemented.
5. Print an empty tree - Empty input file is handled.
6. Empty lines in input file (keys4.txt) - fails.

Score - 9/20.

*/