Assignment 4

COS 212

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

Department of Computer Science
Deadline: 11/04/2014, 12:00

General instructions:

- This assignment should be completed individually, no group effort is allowed.
- Be ready to upload your assignment well before the deadline as no extension will be granted.
- You may not import any of Java's built-in data structures. Doing so will result in a mark of 0. You may only make use native of arrays where applicable. If you require additional data structures, you will have to implement them yourself.
- If your code does not compile you will be awarded a mark of 0. Only the output of your program will be considered for marks.
- Read the entire assignment before you start coding.
- Your source code itself will not be marked. However, to ensure that you followed the instructions, and did not plagiarize, your code may be inspected.

After completing this assignment:

Upon successful completion of this assignment you will have implemented your own B-Tree.

Your task

Refer to section 7.1 (specifically section 7.1.1) in the textbook. You will have to implement your own B-Tree as described in the relevant sections. Your B-Tree should contain `Integer` objects and should allow for `m` to be specified during object creation.

Download the archive `Assignment4.zip` from the course website. This archive contains partial code which you must complete in order to create a B-Tree. Each method is described in the comments within the partial code given to you. You have to write code to test your implementation in the `Main.java` file.

Your tree should allow for insertions and deletions as discussed in the textbook. You may add any additional helper methods that you require but keep in mind that you must still produce a functioning B-Tree.

Some implementation details:

- To delete from the tree, delete by copying is always applied. Use the direct predecessor (i.e. the largest key smaller than the key that you would like to delete) when applying delete operations from internal nodes.
- When a sibling needs to participate in an operation, especially for deletes, always consider the left sibling first before considering the right sibling.

Examples:

Consider the B-Tree in *figure 1*, assuming all insert and delete operations have been completed up until now. For this tree, $M = 5$. For the assignment, you will have to cater for any positive uneven M

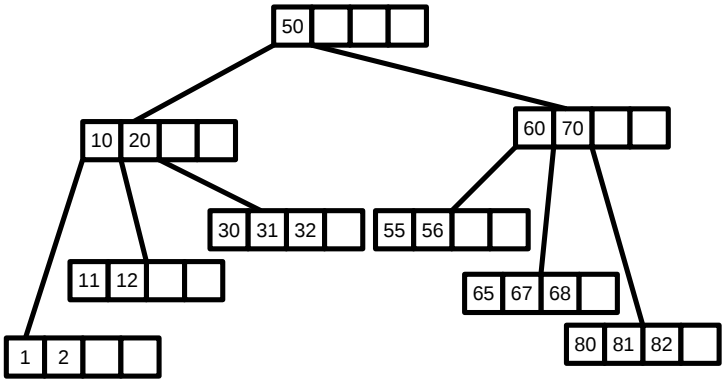

Figure 1: B-Tree

Nodes as Strings

A number of the methods in the `BTree` class will require that nodes be represented as strings. The string representation of the node containing the keys 65, 67 and 68 is:

```
[65] [67] [68] []
```

The string representation of the root node would be:

```
[50] [] [] []
```

You need to have as many sets of square brackets as the number of keys that your nodes can contain, not just the number of keys it currently contains. The elements in the node should appear in ascending order from left to right. Empty keys are indicated by a matching set of square brackets with no spaces between them. There should be no white space in the string representations of your nodes.

Searching

You will be required to return a string representation of all the nodes considered during a searching operation. The nodes should be comma separated with no white space. In the tree in figure 1, searching for the element 67 should yield the string:

```
[50][][][],[60][70][][],[65][67][68][]
```

A null node, where applicable, is indicated with the string:

```
*NULL*
```

Null nodes are used to indicate that the element searched for does not appear in the tree. Therefore, searching for the element 100 should yield the string:

```
[50][][][],[60][70][][],[80][81][82][],*NULL*
```

Once again, you should not have any white space in your strings.

Compile, Run and Submission Instructions:

Your `Main.java` file will be replaced for marking purposes. Your code should compile and run with the following commands executed in any directory:

```
javac *.java
java Main
```

Once you are satisfied that everything is working, you must tar all of your Java code into one archive called sXXX.zip, where XXX is your student/staff number. Submit your code for marking under the appropriate link (Assignment 4) before the deadline.