Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 4 REPORT

EBRU KARDAŞ
141044049

Course Assistant: Burak Koca

1 INTRODUCTION

1.1 Problem Definition

Part 1: Family tree is wanted with “general tree structure”.
Part 2: A general search tree structure where the tree includes multidimensional items. Each level of a mds tree (multidimensional search tree) splits all children along a specific dimension, using a hyperplane that is perpendicular to the corresponding axis.

1.2 System Requirements

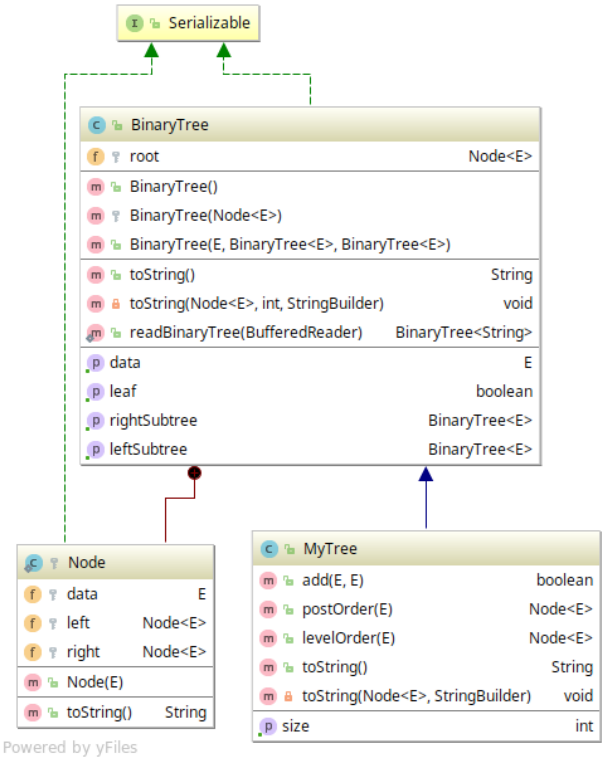
Part 1:
Class is extended from BinaryTree. Methods in BinaryTree can be used directly.
- **Add** method that takes a parentItem and a childItem and inserts the childItem as the last child of the parentItem if the parentItem is already in suach a tree structure. The method returns true if insertion is successful and false if the parentItem is not in the tree.Search method: **levelOrderSearch** () traverses the tree in level order to find ‘argument target’; first the root node (i.e., the node in the first level), then the children of the root node (i.e., nodes in the second level), then the children of the nodes in the second level (i.e., nodes in the third level), and so on. **postOrderSearch** () traverse a node to find ‘argument target’ before traversing its subtrees starting from the root node. Search for the item during traversal. Return the Node reference if the item is in the tree and null if it is not in the tree.

Part 2:
Multi dimensional item tree extends BinaryTree and implements SearchTree.
Each level of a mds tree (multidimensional search tree) splits all children along a specific dimension, using a hyperplane that is perpendicular to the corresponding axis. At the root, all children will be split based on the first dimension (i.e. if the first-dimension coordinate is less than the root it will be in the left-subtree and if it is greater than the root it will be in the right-subtree). Each level down in the tree divides on the next dimension, returning to the first dimension once all others have been exhausted. So, every internal node can be thought of as implicitly generating a splitting hyperplane that divides the space into two parts, half-spaces. The items to the left of this hyperplane are represented by the left-subtree of that node and items right of the hyperplane are represented by the right subtree.

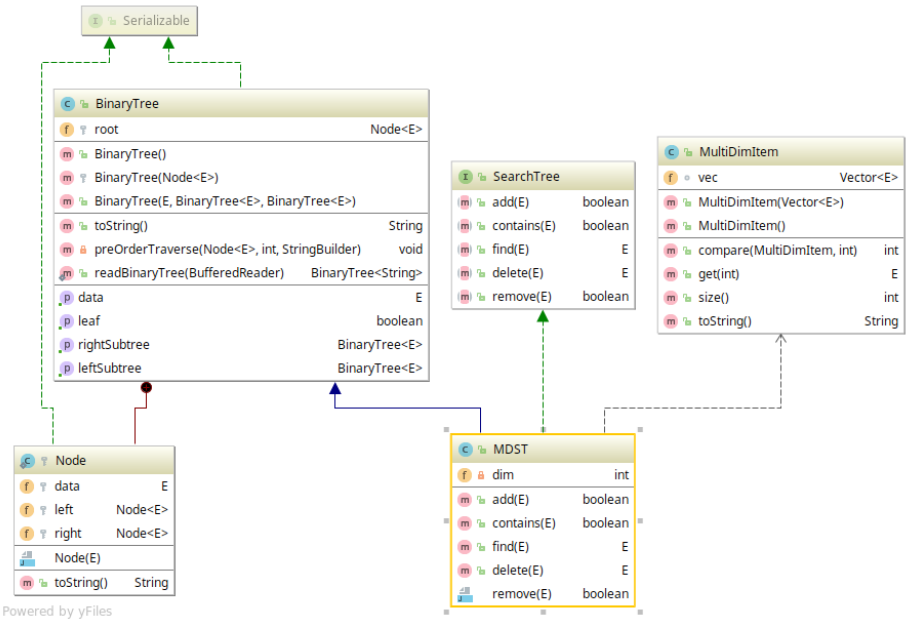
2 METHOD

2.1 Class Diagrams

Part1:

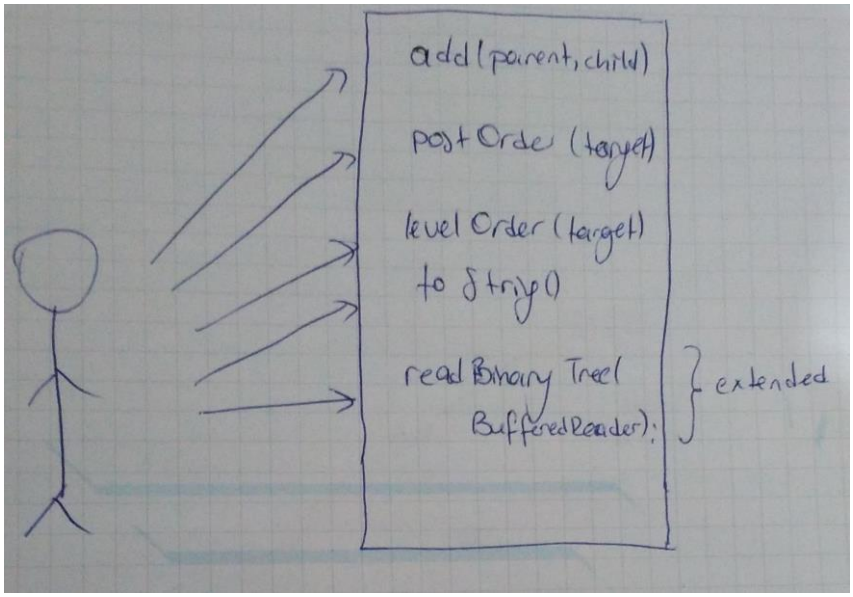


Part2:



2.2 Use Case Diagrams

Part1:



Part2:

2.3 Other Diagrams (optional)

No need other diagrams

2.4 Problem Solution Approach

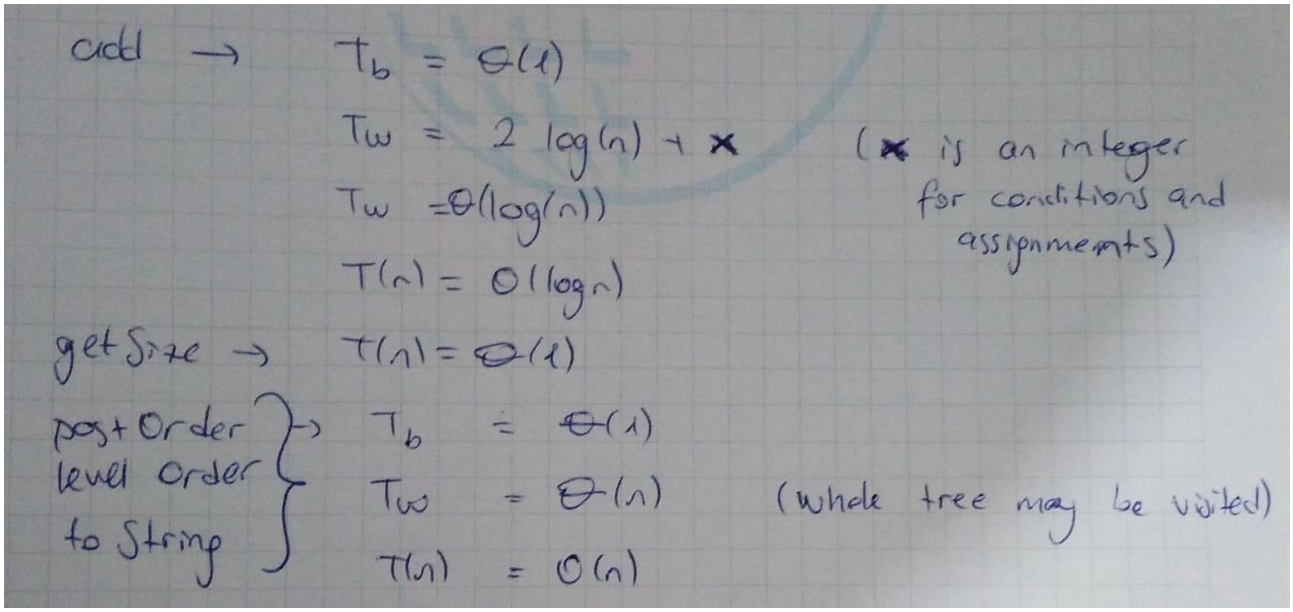
Part 1:

Wanted features are supported. Tree implemented as a “general” family tree.

Left node is first child and all right nodes are siblings.

Search methods wiht level order and post order traverses are iterative and used LinkedList to support traverse structure.

ToString method has a helper recursive method and StringBuilder is used to support recursion.



Part2:

To suppor dimension, I choose MultiDimItem type and it holds vector for dimensions. All methods are overridden but delete method is not completed.

3 RESULT

3.1 Test Cases

Part 1:

A test construct two separate trees by adding multiple new items in the tree.

Test is in Main.java file as test().

Part 2:

Test is in Main.java

3.1 Running Results

Part1:

Outputs are in output folder (folder 1). Files are .txt form.

Part2:

Outputs are in output folder (folder 1). Files are .txt form.