

GIT Department of Computer Engineering
CSE 222/505 - Spring 2018
Homework 04

Part 1-

Consider the binary tree representation of general trees in our textbook where the left branch of a node is the first child, and each right branch is connected to the next sibling (if any).

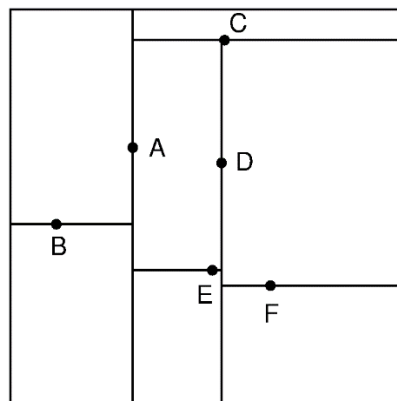
- Extend the BinaryTree class to implement general trees.
- Write an add method that takes a parentItem and a childItem and inserts the childItem as the last child of the parentItem if the parentItem is already in such a tree structure. The method returns true if insertion is successful and false if the parentItem is not in the tree.
- Write the following methods to search an item in such a general tree structure:
 - a) levelOrderSearch (): Traverses the tree in level order; 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.
 - b) postOrderSearch (): Traverse a node 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.
- Override the preOrderTraverse method to traverse to obtain the string representation of the tree.
- As a test construct at least two separate trees by adding 12 new items in the tree. You should try to cover all possible cases with these insertions. You should check the resulting tree by converting it into a string after each insertion.
- Analyze time complexity of your methods

Note that, method representations are not the prototypes of methods.

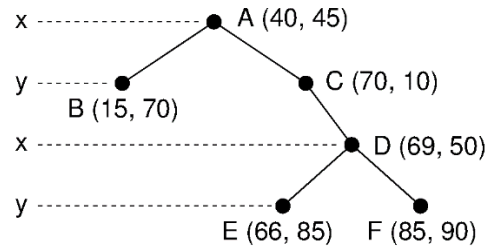
Part 2-

Construct 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. 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. For example, in a 3-dimensional tree, the root would have an x-aligned plane, the root's

children would both have *y*-aligned planes, the root's grandchildren would all have *z*-aligned planes, the root's great-grandchildren would all have *x*-aligned planes, the root's great-great-grandchildren would all have *y*-aligned planes, and so on.



(a)



(b)

Extends the `BinaryTree` class to implement mds trees. Your class should implement the `SearchTree` interface.

Note:

- Obey OOP principles
- Use meaningful and related class, variable, method etc. names
- Use IntelliJ IDE on the given VM. VM download link can be found on moodle(in HW1)
- Your submission is HW04_studentnumber.zip and include following files:
 - o IntelliJ project file
 - o Report.pdf
 - o Javadoc
- The report must be in format "ReportFormat.doc" which was used in HW1
- The implementations will be 75 points and the report is 25 points out of 100
- Submit your homework until the last submission date
- For your questions about homework, feel free to send an email b.koca@gtu.edu.tr

Good Luck!