- Don’t forget to set your Eclipse workspace and working set.
- You must submit the JAR file, exported (with source code), from your Eclipse project.
- You must check your JAR file to make sure all the source files (.java files) are present. It can be opened with file compression programs such as 7-zip or Winrar.
- Failure to export properly will result in your work not getting marked.

1) To submit:
    - Export your project to a JAR file, with source code.
    - Name your JAR file ID_Week10_Q1.jar. For example, 6623110021_Week10_Q1.jar
    - Submit the JAR file on MyCourseville.

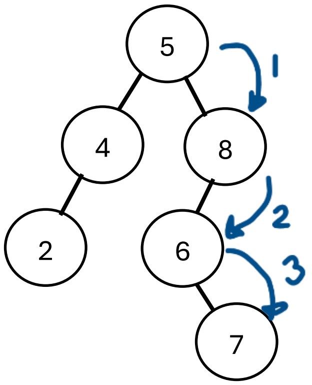
Create eclipse project

You are writing code for Binary Search Tree.

For **the given class, BST**, write the following methods:

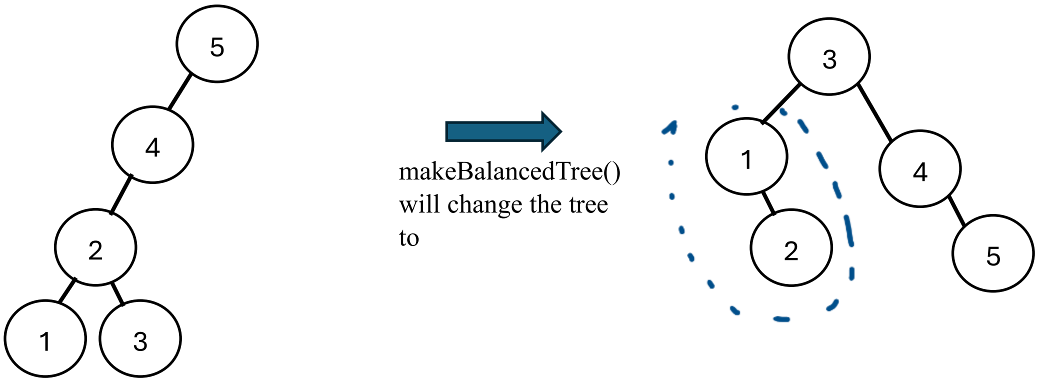
- public int height();
    - calculate the height of the tree. It is the number of branches **furthest** away from the root.

For example, the following tree has height =3.

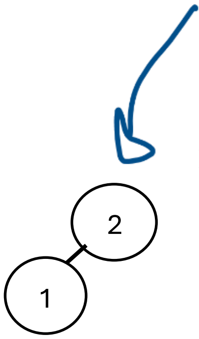


- public void makeBalancedTree():
    - This method reads the content of this BST and creates a new BST (overwriting the original tree).
    - The changed tree distributes nodes evenly on its left and right subtrees (so as each subtree).

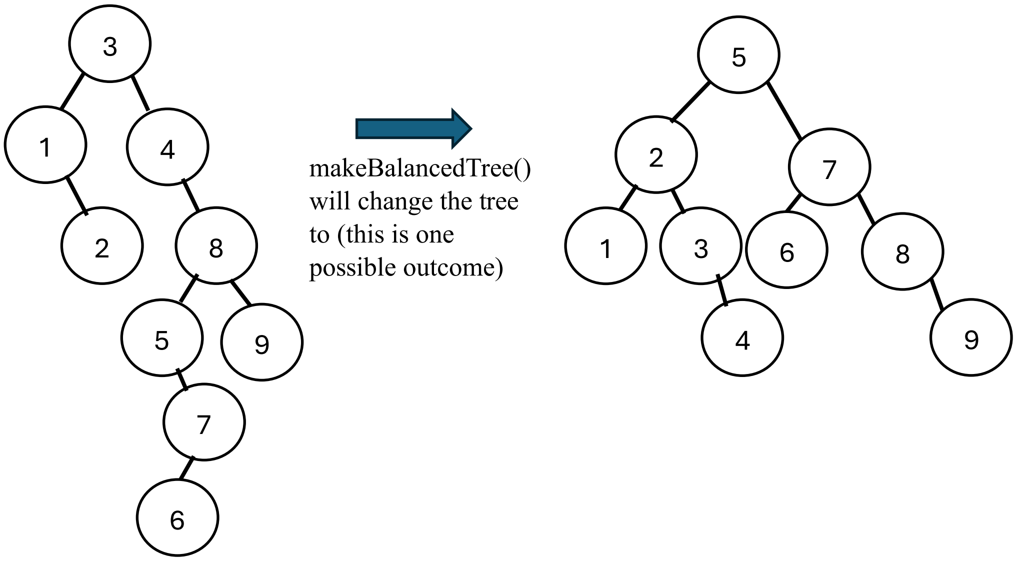
For example, if our original BST looks like:



makeBalancedTree()
will change the tree
to



Each resulting subtree shape can vary, as long as its left and right subtrees have the difference in number of nodes equals to 0 or 1. This is another possible shape of the above subtree.



makeBalancedTree() will change the tree to (this is one possible outcome)

Important Hints:

1. Trees that contain the same number sequence (but different tree shape) can change to the same shape after executing makeBalancedTree().

2. Writing a method that creates an array that stores sorted data from the tree may help. Don't forget to deal with the case where the root of the tree is null.

3. Some methods are already implemented (you don't need to use them, they are used by test cases):

- `public boolean contains(int value)`: tests whether the tree stores the given value.
- `public int numNodes(BSTNode n)`: number of nodes in subtree which has n as its root.
- `public boolean isBalanced(BSTNode n)`: tests whether a subtree with n as its root is actually a balanced tree.

4. You can write your own methods.

Scoring Criteria:

You must only modify BST.java. Other files will not be marked!

The total score is 20 (will be scaled to equal to any other homework).

Run the given JUnit file (TestBST) (If you do not write your code, you will not get any marks):

- `testheight                    2 marks`

Testheight2	2 marks
testmakeBalancedTree1	1 mark
testmakeBalancedTree2	2 marks
testmakeBalancedTree3	4 marks
testmakeBalancedTree4	9 marks

You can create new array or ArrayList in this question.