

Lab 7

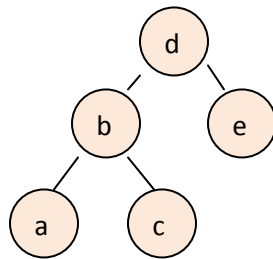
Objectives:

- Learn to build a binary search tree
- Learn to traverse the tree in different ways
- Submit one file: *SBinaryST.java* at the end of the lab. Click the “Submit” button

According to your textbook on page 594, “each node n in a binary search tree satisfies the following three properties:

- n ’s value is greater than all values in its left subtree T_L
- n ’s value is smaller than all values in its right subtree T_R
- both T_L and T_R are binary search trees ”

In this lab, we are going to implement a simple string binary search tree called *SBinaryST*. The “insert” method inserts a string into the tree and keep the property above. There are four ways to traverse the tree: in-order, pre-order, post-order and level-order. For example, if the tree looks like this:



To traverse the tree in the four different orders generates the following sequences:

In-order: a b c d e

Pre-order: d b a c e

Post-order: a c b e d

Level order: d b e a c

It is optional if you could print the sequences in the following format:

In-order: {a, b, c, d, e}

Pre-order: {d, b, a, c, e}

Post-order: {a, c, b, e, d}

Level order: {d, b, e, a, c}

It is optional if you could print the shape of the tree (tilted 90 degrees):

```
e
d
 c
b
 a
```

Download the java files and implement the SBinaryST.java.

Use Connex to submit one file: *SBinaryST.java* at the end of the lab. Make sure you click the “Submit” button.