# Lab 5: Exploring Binary Search Trees (BSTs)

## Objective

In this lab, you'll build a Binary Search Tree (BST) from random numbers and measure its height and imbalance.

## What You’ll Do

### Part 1: Build a Binary Search Tree

- Use random.sample() in Python to generate a random order of numbers from 1 to 20.  
- Insert each number into a Binary Search Tree (BST) using the provided insert() function.

### Part 2: Measure Tree Height

- Use the height() function to calculate the height of the tree after it's built.  
- Repeat this for 10–20 trees to observe differences in tree height.

### Part 3: Measure Tree Imbalance

- Use the imbalance() function to calculate the difference in height between the left and right subtrees of the root.  
- Record the imbalance for each tree in a list or table.

## What to Submit

1. Python script that:  
- Builds and inserts nodes into BSTs  
- Measures height and imbalance  
- Prints your results (no graphs or histograms required)  
  
2. A short report (1 page is fine) that:  
- Briefly explains how your BST works  
- Includes example output of height and imbalance  
- Explains what you noticed across different trees

## Helpful Tips

- Use the random module to generate shuffled sequences  
- Keep your code simple—focus on understanding how BSTs grow and how imbalance happens  
- Use print statements to show your height and imbalance data