



FACULTY OF COMPUTER SCIENCE AND ENGINEERING
Ghulam Ishaq Khan Institute of Engineering Sciences and
Technology, Topi

Lab Duration: 3 hr.

CS351L Intro to AI Lab

Lab No: 3

Instructor: Memoona Saleem

Task Statement:

In this task, you will implement Breadth-First Search (BFS) and Depth-First Search (DFS) algorithms on a binary tree. You will then visualize the tree using both the `'binarytree'` and `'anytree'` libraries. Finally, you will calculate the time complexity for each algorithm and plot the results using Matplotlib.

Instructions:

1. Implement BFS and DFS Algorithms:

- Write Python functions to perform BFS and DFS traversal on a binary tree.
- You can use any binary tree representation of your choice, such as a custom class or the `'binarytree'` library.

2. Visualize the Binary Tree:

- Utilize the `'binarytree'` library to generate a binary tree.
- Use the `'anytree'` library to create an equivalent tree structure for visualization purposes.

3. Apply BFS and DFS Algorithms:

- Apply the BFS and DFS algorithms implemented in step 1 on the binary tree generated in step 2.
- Record the traversal path for each algorithm.

4. Calculate Time Complexity:

- Measure the execution time of BFS and DFS algorithms using Python's `'time'` module.
- Vary the size of the input tree (number of nodes) to analyze the scalability of the algorithms.
- Record the time taken for each algorithm for different tree sizes.

5. Plot Time Complexity Results:

- Use Matplotlib to plot a graph showing the time complexity (execution time) of BFS and DFS algorithms against the size of the input tree