National University of Computer & Emerging Sciences

AL2002 – Artificial Intelligence – Lab (Spring 2025)

BSCS-6B

Lab Work 3 (Breadth First Search, Depth First Search, Depth Limited Search, Iterative Deepening Search, Uniform Cost Search, Bidirectional Search)

Lab Instructor	Momna Javaid
Department	Computer Science

Instructions:

Note: Carefully read the following instructions

1. You also have to submit .ipynb file.
2. Plagiarism is strictly prohibited, 0 marks would be given to students who cheat.
3. Write codes in PYTHON language and you may use any IDE or Notebook environment.
4. First think about the problem statements before starting your programming.
5. At the end when you have done your tasks, attached .py or .ipynb files on google classroom.
6. Paste your complete code in word file along with output in case of .py file (Make sure your submission is completed). In case of missing any file, marks will be deducted.
7. Please submit your file in this format 22Fxxxx_Name_SecB_Lab#
8. Do not submit your assignment after deadline. Late and email submission is not accepted.
9. Do not copy code from any source otherwise you will be penalized with ZERO Marks.
10. YOUR MARKING WILL BE BASED ON THE PRIOR SUBMISSION OF YOUR CODE BEFORE DEADLINE.

Lab Tasks:

Task 1:

This is the representation of a real-world web crawling graph in Figure 1. The homepage has links to the Products, Blogs, About Us, and Contact Us pages. The Products page has links to individual product pages. The Blogs page has links to individual blog posts. Use Breadth First Search to traverse this graph. Calculate the time complexity and space complexity of this problem. Display the complete path upon reaching a goal state.

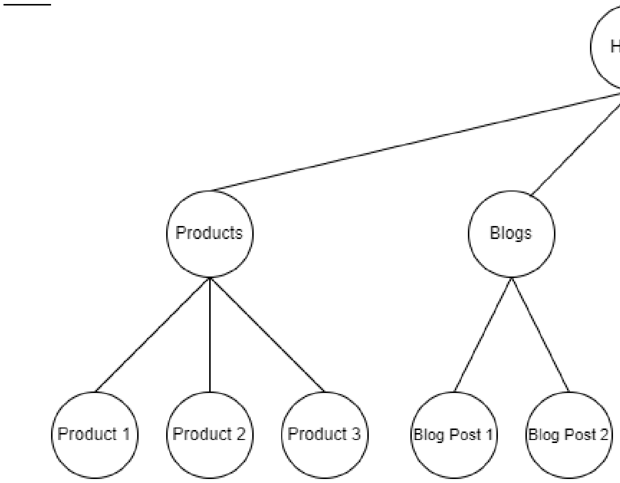

Figure 1. Web Page Structure

Task 2:

Implement Task 1 with Depth First Search. Calculate the time complexity and space complexity of this problem. Display the complete path upon reaching a goal state.

Task 3:

Use the graph from the file EnglishDictionary.txt. Your task is to create an Depth Limited Search so that a user can use this digital dictionary to find the synonyms for a word. Display the path to reach the word searched by the user in IDE output as well as in a separate file named “SearchResult.txt”.

Also find out the time complexity and space complexity of this problem. Display the complete path upon reaching a goal state. Here goal is the searched word. Take input of the graph using file handling. Take input from the user for depth in DLS.

Task 4:

Implement Task 3 with Iterative Deepening Depth Fist Search. Calculate the time complexity and space complexity of this problem. Display the complete path upon reaching a goal state. Here goal is the searched word. Take input of the graph using file handling.

Task 5:

```
graph = {Faisalabad: [(Lahore, 2), (Chiniot, 1), (Islamabad, 4), (Sargodha, 2)],
         Lahore: [(Islamabad, 5), (Faisalabad, 2)],
         Chiniot: [(Islamabad, 6), (Lahore, 3)],
         Rawalpindi: [(Islamabad, 1), (Murree, 1)]}
```

Use this graph to do Uniform Cost Search. Calculate time complexity, Space Complexity and path to reach goal.

Task 6:

Implement Task 1 with Bidirectional Search. Calculate time complexity, Space Complexity and path to reach goal.