



## NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES FAST - PESHAWAR CAMPUS

**Subject:** AL 2002 - Artificial Intelligence Lab  
**Instructor:** Muhammad Saood Sarwar

### Lab Task: Uninformed Searches

## Question 1: Depth-First Search for File Search

**Problem Statement:** Implement a program that uses Depth-First Search (DFS) to search for a file with a given name in a directory tree and returns a list of full paths to all occurrences of the file. If the file is not found, return a list containing the string “File not found.”

#### Input:

- A dictionary representing the directory tree (with directories and files).
- The file name to search for.

#### Output:

- A list of strings with full paths to matching files.
- If no matches are found, return [“File not found.”].

**Example:** For a directory tree with the file “target\_file.txt” in multiple locations, the output might be:

```
['root/dir1/subdir1/target_file.txt', 'root/dir2/target_file.txt']
```

#### Sample Dataset:

```
directory_tree = {
    'root': {
        'files': ['file1.txt', 'file2.txt'],
        'directories': {
            'dir1': {
                'files': ['file3.txt'],
                'directories': {
                    'subdir1': {
                        'files': ['target_file.txt'],
                        'directories': {}
                    }
                }
            }
        }
    }
}
```

```
        }
      }
    },
    'dir2': {
      'files': ['file4.txt', 'target_file.txt'],
      'directories': {
        'subdir2': {
          'files': ['file5.txt'],
          'directories': {}
        }
      }
    }
  }
}
```

**Additional Notes:**

- The search is case-sensitive.
- Paths are constructed using forward slashes (/) to separate directories and files.

## Question 2: Breadth-First Search for Social Network Recommendations

**Problem Statement:** Design a program that recommends users who are two degrees away from a given user in a social network using BFS. The system should:

- Represent the social network as a graph where users are nodes and friendships are edges.
- Use BFS to find all users exactly two degrees away (friends of friends) but exclude direct friends and the user themselves.

**Sample Dataset:**

```
social_network = {
  'Ahmed': ['Bilal', 'Chaudhry'],
  'Bilal': ['Ahmed', 'Danish', 'Emaan'],
  'Chaudhry': ['Ahmed', 'Farhan'],
  'Danish': ['Bilal', 'Gulzar'],
  'Emaan': ['Bilal'],
  'Farhan': ['Chaudhry'],
  'Gulzar': ['Danish']
}
```

**Example:** For the given user “Ahmed”, the expected output should be:

```
['Danish', 'Emaan', 'Farhan']
```

**Additional Notes:**

- The search is case-sensitive.