Fr. Conceicao Rodrigues College of Engineering
Fr. Agnel Ashram, Bandstand, Bandra (W), Mumbai - 400050

Department of Computer Engineering
Academic Term II: 23-24

Class: B.E (Computer), Sem – VI **Subject Name: Artificial Intelligence**

Student Name: Pushpendersingh Bisht **Roll No:9526**

Practical No:	**3**
Title:	Use DFS problem solving method for a) Water Jug Problem b) Missionaries & Cannibals
Date of Performance:	12/02/2024
Date of Submission:	19/02/2024

Rubrics for Evaluation:

Sr. No	Performance Indicator	Excellent	Good	Below Average	Marks
1	On time Completion & Submission (01)	01 (On Time)	NA	00 (Not on Time)	
2	Logic/Algorithm Complexity analysis (03)	03(Correct)	02(Partial)	01 (Tried)	
3	Coding Standards (03): Comments/indention/Naming conventions Test Cases /Output	03(All used)	02 (Partial)	01 (rarely followed)	
4	Post Lab Assignment (03)	03(done well)	2 (Partially Correct)	1(submitte d)	
Total					

Signature of the Teacher:

Experiment No: 3

Title: Use DFS problem solving method for

a) Water Jug Problem
b) Missionaries & Cannibals

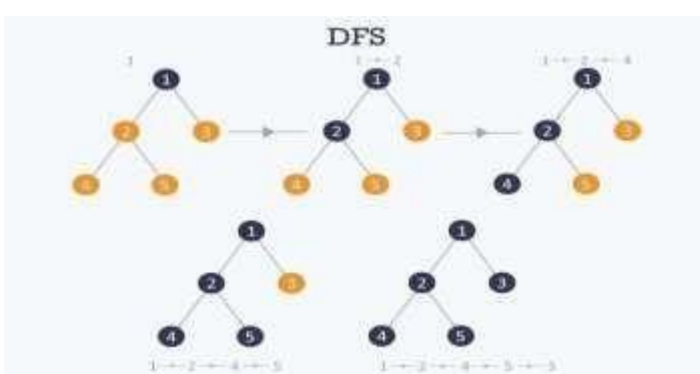
Objective: To write programs which solve the water jug problem and Missionaries & Cannibals problem in an efficient manner using Depth First Search.

Theory:

Depth-first search (DFS) is an algorithm for searching a graph or tree data structure. The algorithm starts at the root (top) node of a tree and goes as far as it can down a given branch (path), then backtracks until it finds an unexplored path, and then explores it. The algorithm does this until the entire graph has been explored.

Depth first search is another way of traversing graphs, which is closely related to preorder traversal of a tree. Recall that preorder traversal simply visits each node before its children. It is most easy to program as a recursive routine:

```
preorder (node v)
{
visit(v);
for each child w of v
preorder(w);
}
```



a) WATER JUG PROBLEM

Given a 'm' liter jug and a 'n' liter jug, both the jugs are initially empty. The jugs don't have markings to allow measuring smaller quantities. You have to use the jugs to measure d liters of water where d is less than n.
(X, Y) corresponds to a state where X refers to amount of water in Jug1 and Y refers to amount of water in Jug2
Determine the path from initial state (xi, yi) to final state (xf, yf), where (xi, yi) is (0, 0) which indicates both Jugs are initially empty and (xf, yf) indicates a state which could be (0, d) or (d, 0).

The operations you can perform are:

1. Empty a Jug, (X, Y)-> (0, Y) Empty Jug 1
2. Fill a Jug, (0, 0)-> (X, 0) Fill Jug 1
3. Pour water from one jug to the other until one of the jugs is either empty or full, (X, Y) -> (X-d, Y+d)

Just like we did for BFS, we can use DFS to classify the edges of G into types. Either an edge vw is in the DFS tree itself, v is an ancestor of w, or w is an ancestor of v. (These last two cases should be thought of as a single type, since they only differ by what order we look at the vertices in.) What this means is that if v and w are in different subtrees of v, we can't have an edge from v to w. This is because if such an edge existed and (say) v were visited first, then the only way we would avoid adding vw to the DFS tree would be if w were visited during one of the recursive calls from v, but then v would be an ancestor of w.

Post Lab Assignment:

1. What is the time complexity of the Water Jug problem?
2. Why is DFS not used for solving a water jug problem?

9526 TE COMPS A A.I Exp3

Postlab:-

① What is the time complexity of the water jug problem?
=> The time complexity of the Water Jug problem, using BFS or DFS is approximately $O(m \times n)$. where m and n are the capacities of the two jugs.

② Why is DFS not used for solving a water jug problem?
=> DFS is not typically used for solving the Water jug problem because it may get stuck in deep branches without finding a solution, does not guarantee finding the shortest path to the solution, and may exhaust memory if the search space is large.