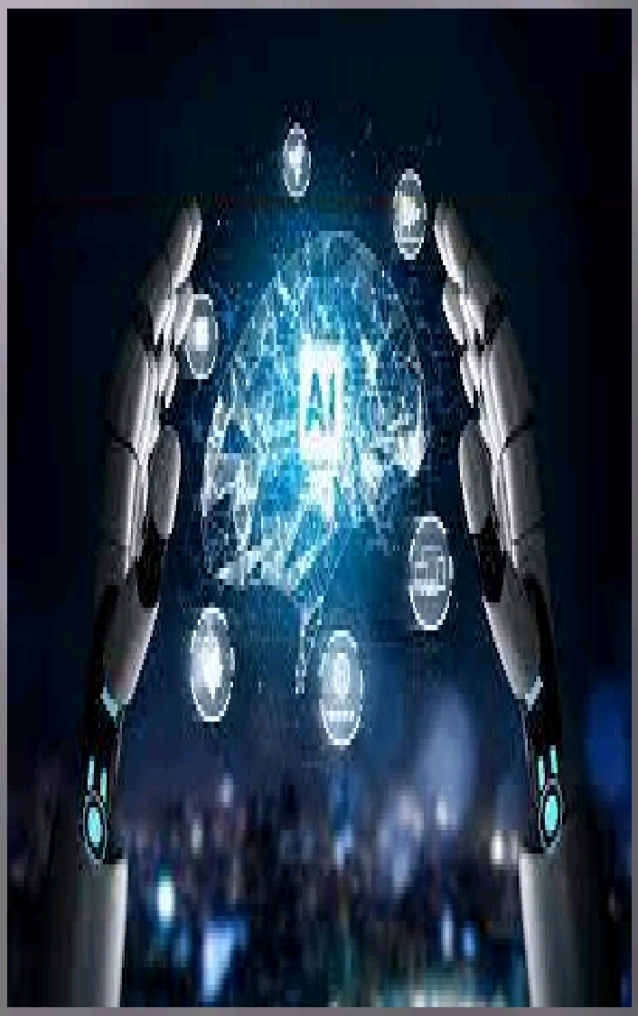
ARTIFICIAL INTELLIGENCE LAB MANUAL



R.SIVARANJANI M.E

(Regulation 2021)

ISBN 978-93-6128-893-7

ARTIFICIAL INTELLIGENCE LAB MANUAL

(COMPUTER SCIENCE AND ENGINEERING)

REGULATION – 2021

AUTHOR:

Mrs.R.SIVARANJANI.,M.E.,

GENERAL GUIDELINES AND SAFETY NSTRUCTIONS

1. Sign in the log register as soon as you enter the lab and strictly observe your lab timings.
2. Strictly follow the written and verbal instructions given by the teacher / Lab Instructor. If you do not understand the instructions, the handouts and the procedures, ask the instructor or teacher.
3. **Never work alone!** You should be accompanied by your laboratory partner and / or the instructors / teaching assistants all the time.
4. It is mandatory to come to lab in a formal dress and wear your ID cards.
5. Do not wear loose-fitting clothing or jewels in the lab. Rings and necklaces are usual excellent conductors of electricity.
6. Mobile phones should be switched off in the lab. Keep bags in the bag rack.
7. Keep the labs clean at all times, no food and drinks allowed inside the lab.
8. Intentional misconduct will lead to expulsion from the lab.
9. Do not handle any system without reading the safety instructions. Read the handout and procedures in the Lab Manual before starting the experiments.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

REGULATION – 2021

AD3311-ARTIFICIAL INTELLIGENCE LABORATORY

Mrs.R.SIVARANJANI.,M.E.,

Assistant Professor/ COMPUTER SCIENCE ENGINEERING

Annai Mira College of Engineering and Technology

Ranipet – 632 517

PREFACE

This book on "**ARTIFICIAL INTELLIGENCE LABORATORY MANUAL (COMPUTER SCIENCE Engineering)**" covers the complete syllabus prescribed by the Anna University, Chennai for the fourth semester **B.E/ B.Tech**. Degree course under **Outcome Based Education Credit System with the new regulation 2021**.

This book covers : 8-PUZZLE,8-QUEEN PROBLEMS,CRYPTARITHMETIC,A*ALGORITHMS,MINIMAX ALGORITHMS,CSP,FORWARD AND BACKWARD CHAINING.

We hope that this book will be useful to both teachers and students. Finally we would request the readers to kindly send their valuable comments and suggestions towards the improvement of the manual and the same will be gratefully acknowledge.

Any suggestion from the reader for the betterment of this book can be dropped into sivaranjani271996@gmail.com.

Mrs.R.Sivaranjani.,M.E.,

ACKNOWLEDGEMENT

We are thankful to and fortunate enough to get constant encouragement, support and guideline from Chairman **Thiru.S.Ramadoss Ayya**, Secretary & Treasurer **Mr.G.Thamotharan** for his blessings to complete the book successfully.

We would not forget to remember our Principal **Dr.T.K.Gopinathan** for his constant assistance in preparing this book.

ANNAI MIRA COLLEGE OF ENGINEERING AND TECHNOLOGY

NH-46, Chennai-Bengaluru National Highways, Arapakkam,

Vellore-632517, TamilNadu, India

Telephone: 04172-292925 Fax: 04172-292926

Email:amcet.rtet@gmail.com/info@amcet.in Web: www.amcet.in

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

AD3311 - ARTIFICIAL INTELLIGENCE LABORATORY

Name : ……..

Register Number : ……

Year & Branch : ……..

Semester : ……

Academic Year : ……

NH-46, Chennai-Bengaluru National Highways, Arappakkam,
Vellore-632517, TamilNadu, India
Telephone: 04172-292925 Fax: 04172-292926

CERTIFICATE

This is to certify that the bonafide record of the practical work done by …… Register Number ……..…..……... of II year B.Tech (Artificial Intelligence and Data Science) submitted for the B.Tech - Degree practical examination **(III Semester)** in **AD3311 - ARTIFICIAL INTELLIGENCE LABORATORY** during the academic year 2023 -2024

Staff in -Charge **Head of the Department**

Submitted for the practical examination held on ----------

Internal Examiner **External Examiner**

TABLE OF CONTENTS

Exp No: 1(a)	DATE:__/__/20__

IMPLEMENT BASIC SEARCH STRATEGIES—8-PUZZLE PROBLEM

AIM:

To implement basic search strategies –8-Puzzle Problem.

ALGORITHM:

1. The code starts by creating a Solution class and then defining the method solve.
2. The function takes in a board as an argument, which is a list of tuples representing the positions on the board.
3. It iterates through each position in the list and creates a dictionary with the position's value set to 0.
4. Then it iterates through all possible moves for that positon and returns their number of occurrences in dict.
5. After their, it loops over all nodes on the board until it finds one where there are no more moves left to make(i.e., when len(current_nodes)==0).
6. This node will be returned as -1 if found or else its index will be stored into pos_0 so that we can find out what move was made at that point later on.
7. The next step is finding out what move was made at every node by looping over all possible moves for each node using self's find_next function, which takes in a single node as an argument and returns any other nodes connected to it via path-finding algorithms like DFS or BFS(see below).
8. For example, if pos_0=1 then self would call: moves={0:[1],1:
9. The code will print the number of paths in a solution.

PROGRAM:

```
class Solution:

  def solve(self, board):

    flatten = tuple(board[0] + board[1] + board[2])

    dict = {flatten: 0}

    if flatten == (0, 1, 2, 3, 4, 5, 6, 7, 8):

      return 0

    return self.get_paths(dict)
```

```
def get_paths(self, dict):
    cnt = 0
    while True:
        current_states = [state for state, distance in dict.items() if distance == cnt]
        if len(current_states) == 0:
            return -1
        for state in current_states:
            next_states = self.find_next(state)
            for next_state in next_states:
                if next_state not in dict:
                    dict[next_state] = cnt + 1
                if next_state == (0, 1, 2, 3, 4, 5, 6, 7, 8):
                    return cnt + 1
        cnt += 1
def find_next(self, state):
    moves = {
        0: [1, 3],
        1: [0, 2, 4],
        2: [1, 5],
        3: [0, 4, 6],
        4: [1, 3, 5, 7],
        5: [2, 4, 8],
        6: [3, 7],
        7: [4, 6, 8],
        8: [5, 7],
    }
```