



**EASTERN MEDITERRANEAN UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT**

AING216 Basic Search Strategies

LAB II

Experiment: Implementation of A* Algorithm

1. Objective

To implement the A* (A-Star) search algorithm in Python to find the shortest path in:

1. A weighted graph
2. A maze (grid environment)

2. Introduction

The A* algorithm is a popular pathfinding and graph traversal algorithm used in Artificial Intelligence. It combines the advantages of Dijkstra's Algorithm and Greedy Best-First Search. The algorithm evaluates nodes using the function:

$$f(n) = g(n) + h(n)$$

Where:

- $g(n)$ = cost from start node to the current node
- $h(n)$ = heuristic estimate from current node to goal
- $f(n)$ = estimated total cost of the path through node n

3. Software Requirements

- Python 3.x
- Any Python IDE (VS Code / PyCharm / Jupyter Notebook)
- Basic knowledge of data structures and graphs

Part A: A* Algorithm for Graph Search

Problem Statement

Write a Python program to implement the A* algorithm to find the shortest path between a start node and a goal node in a weighted graph.

Graph Structure

A --1--> B
A --3--> C
B --1--> D
B --5--> E
C --2--> F
D --3--> G
E --1--> G
F --2--> G

Heuristic Values

Node	h(n)
A	6
B	4
C	4
D	2
E	1
F	2
G	0

Tasks

1. Represent the graph using Python data structures.
2. Implement the A* search algorithm.
3. Display the shortest path.
4. Print the total path cost.

Expected Output Example

Shortest Path: A → B → D → G
Total Cost: 5

Part B: A* Algorithm for Maze Search

Problem Statement

Write a Python program to find the shortest path in a maze using the A* algorithm.

Maze Representation

The maze is represented as a grid where:

0 = free cell

1 = obstacle/wall

Example Maze

```
S 0 0 0
```

```
1 1 0 1
```

```
0 0 0 0
```

```
0 1 1 G
```

S = Start position (0,0)

G = Goal position (3,3)

Tasks

1. Represent the maze using a 2D list.
2. Use Manhattan Distance heuristic:
$$h(n) = |x1 - x2| + |y1 - y2|$$
3. Implement the A* algorithm to find the path.
4. Print the path coordinates.

Expected Output Example

(0,0) → (0,1) → (0,2) → (1,2) → (2,2) → (2,3) → (3,3)

Deliverables

Students must submit:

- Python source code
- Output screenshots
- Brief explanation of the A* algorithm