Lab 05

A\* Search Alogirhtm

1. Implement A\* Search Using network Scenario

Code

import heapq

railroad\_network = {

    'A': {'B': 4, 'C': 2},

    'B': {'A': 4, 'D': 5, 'E': 10},

    'C': {'A': 2, 'D': 8},

    'D': {'B': 5, 'C': 8, 'E': 2},

    'E': {'B': 10, 'D': 2}

}

heuristic = {

    'A': 10, 'B': 6, 'C': 8, 'D': 3, 'E': 0

}

def a\_star\_search(start, goal):

    open\_set = []

    heapq.heappush(open\_set, (0, start))

    g\_costs = {start: 0}

    came\_from = {start: None}

    while open\_set:

        \_, current = heapq.heappop(open\_set)

        if current == goal:

            path = []

            while current:

                path.append(current)

                current = came\_from[current]

            return path[::-1]

        for neighbor, distance in railroad\_network[current].items():

            new\_cost = g\_costs[current] + distance

            if neighbor not in g\_costs or new\_cost < g\_costs[neighbor]:

                g\_costs[neighbor] = new\_cost

                priority = new\_cost + heuristic[neighbor]

                heapq.heappush(open\_set, (priority, neighbor))

                came\_from[neighbor] = current

    return None

start\_station = 'A'

goal\_station = 'E'

path = a\_star\_search(start\_station, goal\_station)

print("Path from", start\_station, "to", goal\_station, ":", path)

Ouput

