Python program for A \* Algorithm

from queue import PriorityQueue

def a\_star(graph, start, goal, h):

open\_set = PriorityQueue()

open\_set.put((0, start))

came\_from = {}

g\_score = {node: float('inf') for node in graph}

g\_score[start] = 0

while not open\_set.empty():

\_, current = open\_set.get()

if current == goal:

path = []

while current in came\_from:

path.append(current)

current = came\_from[current]

path.append(start)

return path[::-1], g\_score[goal]

for neighbor, cost in graph[current].items():

tentative\_g = g\_score[current] + cost

if tentative\_g < g\_score[neighbor]:

g\_score[neighbor] = tentative\_g

f\_score = tentative\_g + h[neighbor]

came\_from[neighbor] = current

open\_set.put((f\_score, neighbor))

graph = {

'A': {'B':1,'C':4},

'B': {'A':1,'C':2,'D':5},

'C': {'A':4,'B':2,'D':1},

'D': {'B':5,'C':1}

}

heuristic = {'A':7,'B':6,'C':2,'D':0}

path, cost = a\_star(graph,'A','D',heuristic)

print("Shortest Path:", path)

print("Total Cost:", cost)

Output

Shortest Path: ['A', 'C', 'D']

Total Cost: 5