Exp-10 A\* algorithm

Code

import heapq

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

open\_list = []

heapq.heappush(open\_list, (0 + h[start], 0, start, [start]))

visited = set()

while open\_list:

f, g, current, path = heapq.heappop(open\_list)

if current == goal:

return path, g

if current in visited:

continue

visited.add(current)

for neighbor, cost in graph[current]:

if neighbor not in visited:

heapq.heappush(open\_list, (g + cost + h[neighbor], g + cost, neighbor, path + [neighbor]))

return None, float('inf')

graph = {

'A': [('B', 1), ('C', 4)],

'B': [('A', 1), ('D', 2), ('E', 5)],

'C': [('A', 4), ('F', 3)],

'D': [('B', 2)],

'E': [('B', 5), ('F', 1)],

'F': [('C', 3), ('E', 1)]

}

h = {

'A': 7,

'B': 6,

'C': 2,

'D': 3,

'E': 1,

'F': 0

}

start\_node = 'A'

goal\_node = 'F'

path, cost = a\_star(graph, start\_node, goal\_node, h)

print("Path:", path)

print("Total cost:", cost)

output

