from queue import PriorityQueue

graph={

'A': {'B': 1, 'C': 3, 'D': 7},

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

'C': {'D': 12}

}

h\_value={

'A': 1,

'B': 1,

'C': 1,

'D': 1

}

def search(graph, start, end):

queue = PriorityQueue()

queue.put((0+h\_value[start], [start]))

while not queue.empty():

node = queue.get()

current = node[1][len(node[1]) - 1]

if end in node[1]:

print("Path found" ,str(node[1]))

print("cost =" ,str(node[0]))

break

cost = node[0] - h\_value[current]

for neighbor in graph[current]:

temp = node[1][:]

temp.append(neighbor)

queue.put((cost + graph[current][neighbor] + h\_value[neighbor], temp))

search\_city=input("Enter the city to be searched\n")

start\_city=input("Enter the starting city\n")

search(graph, start\_city, search\_city)