import math  
from queue import PriorityQueue  
input\_file = open("21301586\_mirza shafi\_CSE422\_08\_Lab\_Assignment01\_InputFile\_Summer2024.txt", "r")  
data = input\_file.readlines()  
h = {}  
graph = {}  
  
for i in data:  
 if i.strip():  
 parts = i.split(" ")  
 city = parts[0]  
 h[city] = int(parts[1])  
 nei = []  
 for i in range(2, len(parts) - 1, 2):  
 nei.append((parts[i], int(parts[i + 1])))  
 graph[city] = nei  
  
def Astar\_search(graph, start, end, h):  
 queue = PriorityQueue()  
 p = {}  
 d = {}  
 tc = {} #tc = total cost  
 for i in graph.keys():  
 d[i] = math.inf  
 tc[i] = math.inf  
 p[i] = None  
 d[start] = 0  
 tc[start] = h[start]  
 queue.put((tc[start], start))  
 while not queue.empty():  
 current\_c, current\_n = queue.get()  
  
 if current\_n == end:  
 path = []  
 while current\_n is not None:  
 path.append(current\_n)  
 current\_n = p[current\_n]  
 path.reverse()  
 return path, d[end]  
  
 for key, w in graph[current\_n]:  
 new\_distance = d[current\_n] + w  
 total\_costofneighbor = new\_distance + h[key]  
 if new\_distance < d[key]:  
 d[key] = new\_distance  
 p[key] = current\_n  
 queue.put((total\_costofneighbor, key))  
  
 return "NO PATH FOUND", 0  
  
begin = "Arad"  
goal = "Bucharest"  
path, cost = Astar\_search(graph, begin, goal, h)  
if cost != 0:  
 print(f"Path: {'-> '.join(path)}")  
 print(f"Total distance: {cost} km")  
else:  
 print(path)