import heapq

m= {}

n= {}

destin= 'Bucharest'

start= 'Arad'

f = open("input\_file.txt","r")

for i in f:

temp = i.split()

source = temp[0]

destination\_info=[]

m[source]=int(temp[1])

if int(temp[1])==0:

dest=temp[0]

else:

for j in range(2,len(temp),2):

destination\_info.append((temp[j],int(temp[j + 1])))

n[source]= destination\_info

parent= {}

visited= {}

q=[]

heapq.heappush(q, (m[start], start))

visited[start]=0

parent[start]=None

while q:

q= sorted(q, key=lambda x: x[0])

current=q.pop(0)

current\_cost,current\_node=current

if current\_node==dest:

path= []

total\_distance=visited[current\_node]

while current\_node:

path.append(current\_node)

current\_node=parent[current\_node]

path.reverse()

print("Path:", ' -> '.join(path))

print(f"Total distance: {total\_distance} km")

break

for neighbor, cost in n[current\_node]:

#print("a",neighbor,cost)

new\_cost= visited[current\_node]+cost

#print("cost",new\_cost)

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

visited[neighbor]=new\_cost

priority= new\_cost+m[neighbor]

heapq.heappush(q,(priority, neighbor))

parent[neighbor]=current\_node

else:

print("NO PATH FOUND")