def dij(start, graph):

n = len(graph)

costs = [99999 for \_ in range(n)]

costs[start] = 0

path = [-1 for \_ in range(n)]

visited = [False for \_ in range(n)]

t = []

while len(t) < n:

minCost = 99999

minNode = None

for i in range(n):

if not visited[i] and costs[i] < minCost:

minCost = costs[i]

minNode = i

t.append(minNode)

visited[minNode] = True

for edge in graph[minNode]:

if not visited[edge[0]] and minCost + edge[1] < costs[edge[0]]:

costs[edge[0]] = minCost + edge[1]

path[edge[0]] = minNode

return costs, path

data = [

[1, 0, 4],

[1, 2, 5],

[1, 3, 12],

[1, 6, 9],

[2, 0, 1],

[0, 6, 5],

[3, 6, 6],

[3, 4, 8],

[0, 5, 2],

[5, 6, 9],

[5, 3, 8],

[5, 4, 7]

]

n = 7

graph = [[] for \_ in range(n)]

for edge in data:

graph[edge[0]].append([edge[1], edge[2]])

graph[edge[1]].append([edge[0], edge[2]])

costs, path = dij(1, graph)

print("从v1到其他结点:")

print('costs')

print(costs)

print('path')

print(path)

