离散数学实验报告

1. 实验内容

利用电脑计算图中两点间最短路径

1. 代码实现
2. **import** copy  
   **def** Dijkstra(network):  
    passed = [] *# 已经过的点* nopass = [x **for** x **in** range(len(network))] *# 未经过的点* path = [[] **for** x **in** range(len(network))] *# 由0到每个点的最短路径* path\_base = [0] *# 由0到当前位置的最短路径* dis\_base = 0 *# 由0到当前位置的距离* idx = 0 *# 当前位置* dis = network[0] *# 当前位置与各点的边的长度* **while** len(nopass):  
    *# 如果经过新点到目标点的距离更小 则更新路径和距离* **for** i **in** nopass:  
    **if** dis[idx] + network[idx][i] < dis[i] **or** dis[idx] + network[idx][i] == dis[i]:  
    dis[i] = dis[idx] + network[idx][i]  
    path[i] = copy.deepcopy(path[idx])  
    path[i].append(i)  
    *# 寻找与当前位置最近的点* idx=nopass[0]  
    **for** i **in** nopass:  
    **if** dis[i]<dis[idx]:  
    idx = i  
    *# 将该点加入路径中* passed.append(idx)  
    nopass.remove(idx)  
     
    print(dis)  
    print(path)  
     
     
   network = [[0, 2, 999, 5, 999, 999],  
    [2, 0, 3, 1, 7, 999],  
    [999, 3, 0, 1, 3, 5],  
    [2, 1, 1, 0, 2, 999],  
    [999, 7, 3, 2, 0, 1],  
    [999, 999, 5, 999, 1, 0]]  
   Dijkstra(network)