**《运筹学》实验报告**

|  |  |  |
| --- | --- | --- |
| **第 四 次实验** | **日期：11月29日** | **得分：** |
| **学号：** | **姓名：*Steven*** | **专业：智能科学技术** |

1. **实验目的**

理解图的基本概念

熟练使用迪杰斯特拉算法进行最短距离求解

1. **算法设计**

理解分支定界法的数学思路，将其转换为代码进行运算。并将求解结果进行展示

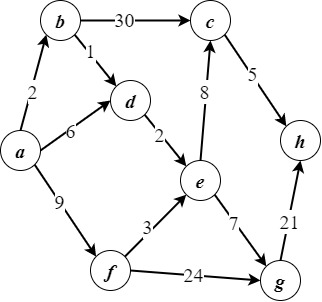
**三、核心程序代码**

|  |
| --- |
| 函数实现 |
| def Dijkstra(G):      """      Dijkstra算法，默认计算index为0的点与其他各点的距离      :param G: 邻接矩阵，不存在的边以及距离为0的边请用99填充      """      n = len(G)  # 计算顶点数量      # v:是否访问      # dis:起始结点到相邻结点的距离      v = [0]\*n      dis = G[0].copy()      # 初始化      v[0] = 1      dis[0] = 99      # 循环n次      for \_ in range(n):          # 找出与集合相邻且距离起点最近的点          k = 0          for j in range(n):              if v[j] == 0 and dis[j] < dis[k]:                  k = j              # 该点被访问          v[k] = 1 # 将该点加入集合          # 用该点进行松弛(relax)          for j in range(n):              if v[j] == 0 and dis[k] + G[k][j] < dis[j]:                  dis[j] = dis[k] + G[k][j]      return dis |
| 函数调用 |
| G = [      [99, 2, 99, 6, 99, 9, 99, 99],      [99, 99, 30, 1, 99, 99, 99, 99],      [99, 99, 99, 99, 99, 99, 99, 5],      [99, 99, 99, 99, 2, 99, 99, 99],      [99, 99, 8, 99, 99, 99, 7, 99],      [99, 99, 99, 99, 3, 99, 24, 99],      [99, 99, 99, 99, 99, 99, 99, 21],      [99, 99, 99, 99, 99, 99, 99, 99]]  print(Dijkstra(G)) |

**四、测试及结果（给出测试用例及测试结果）**

(1) 测试用例：

使用下图作为测试用例：



(2) 运算结果：

文本

描述已自动生成

根据结果显示，该方法找到了结点a与其余各点的距离，由于a与其自身的距离为0，所以显示为99。