

实验报告

课程名称 : 计算机网络原理
实验题目 : 编程实现路由算法
学号 : 21281280
姓名 : 柯劲帆
班级 : 物联网2101班
指导老师 : 常晓琳
报告日期 : 2024年6月9日

目录

目录

1. 实验目的

2. 实验环境

3. 实验过程

3.1. 编写代码

3.2. 运行代码

4. 总结和感想

1. 实验目的

运用各种编程语言实现基于 Dijkstra 算法的路由软件。

2. 实验环境

- OS: WSL2 (内核5.15.146.1-microsoft-standard-WSL2)
- Python: version 3.11.4

3. 实验过程

3.1. 编写代码

```
1 INF = float('inf')
2
3 def Dijkstra(G, origin):
4     node_num = len(G)
5     vis = [False] * node_num
6     dis = [G[origin][node] for node in range(node_num)]
7     paths = [[]] * node_num
8     paths[origin].append(origin)
```

```

9     vis[origin] = True
10    last_point = origin
11    for i in range(node_num - 1):
12        min_dis = INF
13        for j in range(node_num):
14            if not vis[j] and dis[j] < min_dis:
15                min_dis = dis[j]
16                last_point = j
17        vis[last_point] = True
18        if i == 0:
19            paths[last_point] = paths[origin] + [last_point]
20        for k in range(node_num):
21            if G[last_point][k] < INF and dis[k] > dis[last_point] +
G[last_point][k]:
22                dis[k] = dis[last_point] + G[last_point][k]
23                paths[k] = paths[last_point] + [k]
24
25    return dis, paths
26
27
28 if __name__ == '__main__':
29     G = [[0, 1, 12, INF, INF, INF],
30          [INF, 0, 9, 3, INF, INF],
31          [INF, INF, 0, INF, 5, INF],
32          [INF, INF, 4, 0, 13, 15],
33          [INF, INF, INF, INF, 0, 4],
34          [INF, INF, INF, INF, INF, 0]]
35     origin = 0
36     dis, paths = Dijkstra(G, origin)
37     for i in range(len(G)):
38         print(f"{origin} to {i}", "path:", paths[i], f"distance: {dis[i]}")

```

3.2. 运行代码

```

1 $ python main.py
2 0 to 0 path: [0] distance: 0
3 0 to 1 path: [0, 1] distance: 1
4 0 to 2 path: [0, 1, 3, 2] distance: 8
5 0 to 3 path: [0, 1, 3] distance: 4
6 0 to 4 path: [0, 1, 3, 2, 4] distance: 13
7 0 to 5 path: [0, 1, 3, 2, 4, 5] distance: 17

```

代码能够正确求出最短路径和路径长度。

4. 总结和感想

在本次实验中，我通过编程实现了 Dijkstra 算法，用于计算加权图中各节点之间的最短路径。在整个实验过程中，我深入理解了 Dijkstra 算法的核心思想及其在路由算法中的应用。

通过实现 Dijkstra 算法，我对其逐步选取未访问节点中距离最短的节点，并更新周围节点的最短路径的过程有了更深的理解。Dijkstra 算法在处理图中非负权重路径问题时非常高效，能够在多种网络环境下应用。

在实现过程中，我使用 Python 编写了 Dijkstra 算法，并成功运行程序验证了结果的正确性。代码清晰地展示了算法从初始节点出发，通过不断更新距离数组和路径数组来寻找最短路径的过程。

通过本次实验，我深刻体会到 Dijkstra 算法在计算机网络中的重要性。路由算法是网络通信的核心之一，能够高效找到最短路径对于提升网络性能至关重要。

在实验过程中，我意识到代码的优化和效率提升的重要性。尽管 Dijkstra 算法在小规模网络中表现良好，但在大规模网络中，如何进一步优化算法和提高效率是一个值得深入研究的课题。

本次实验不仅增强了我对算法的理解，也提升了我的编程能力。在未来的学习和工作中，我会继续通过实践和动手编程来巩固理论知识，提升实际解决问题的能力。