

BTH004-图的最短路径搜索

姓名：赵水 学号：201806150329

1 问题：

对于给定的一幅图，求从其起点到终点的最短路径

2 算法：

采用DP（动态规划）算法。对于图中的某个节点，若其有连通其他的节点，且从当前节点到下一节点的路径总长度更短，则更新到下一节点的最短路径值。对所有点重复进行此操作，若某一次操作中，所有点的最短路径值都没有更新，则退出循环并输出所求最短路径。

3 代码：

```
import sys

def shortestPath(graph, start, end):
    dp = [sys.maxsize] * len(graph)
    dp[start] = 0

    queue = [] #遍历点顺序，FIFO
    queue.append(start)

    while len(queue) > 0:
        row = queue.pop()
        for i in range(len(dp)):
            if graph[row][i] != 0:
                queue.append(i)
                dp[i] = min(dp[i], dp[row] + graph[row][i])

    return dp[end], dp
```

4 测试

4.1 代码

```
graph = [[0,1,0,3,0],
          [0,0,2,0,5],
          [0,0,0,0,2],
          [0,0,2,0,2],
          [0,0,0,0,0]]
print(shortestPath(graph, 1, 4))
```

4.2 输出

```
(4, [9223372036854775807, 0, 2, 9223372036854775807, 4])
```

#解释:从起点节点1 (row=2) 到终点节点4 (row=5) 的最短路径为4, 节点1到其他点 (列表索引即为节点ID) 的位置如上述链表所示