Computational Thinking and Program Design

Assignment Eleven - Part 2 (10 marks)
(Due on 7 June 2022)

Rocky K. C. Chang

2 June 2022

Instructions:

1. Submit your code for Q1 and `myqueue.py` to the iLearning platform before 23:59 on 7 June.
2. As usual, compress both files into "Your student ID".7z using 7-zip.
3. Late submission will not be accepted.
4. Observe also the penalty for plagiarism as stated in the Course Overview slides.

Question 1 (From BFS algorithm to shortest-path algorithm)

[10 MARKS] In the lecture yesterday, you have seen how we used a backward BFS tree to find a shortest path. In this question you will use a forward BFS tree to find a shortest path. You will implement `findShortestPath()`:

```
def findShortestPath(tree, src, dest):
  # Inputs:
  #   - G: A graph
  #   - src: A source node on G
  #   - dest: A destination node on G
  # Output: Return a shortest path in the form of a list.
```

In the function, you will first call the function for generating a forward BFS tree as you may have done in A12. If you have not done that, you may use my code for A12 from iLearning. After that, you will write additional code to find a shortest path. You cannot import any library other than `myqueue.py`.

Write additional code to test your function for the two graphs below. The expected outputs are given in Figure 1.

```
graph = {
  'A' : ['D', 'E', 'I'],
  'B' : ['D'],
  'C' : ['E', 'H', 'J'],
  'D' : ['A', 'B', 'E', 'J'],
  'E' : ['A', 'C', 'D', 'H'],
  'F' : ['H', 'I'],
  'G' : ['H'],
  'H' : ['C', 'E', 'F', 'G'],
  'I' : ['A', 'F'],
  'J' : ['C', 'D']
```

```
}

MCGW_graph ={
  'EEEE':['WEWE'],
  'WEWE':['EEEE','EEWE'],
  'EEWE':['WEWE','WWWE','WEWW'],
  'WWWE':['EEWE','EWEE'],
  'EWEE':['WWWE','WWEW'],
  'WEWW':['EEWE','EEEW'],
  'EEEW':['WEWW','WWEW'],
  'WWEW':['EWEE','EEEW','EWEW'],
  'EWEW':['WWEW','WWWW'],
  'WWWW':['EWEW']
}
```

```
For the 10-node graph:
A --> B: ['A', 'D', 'B']
A --> C: ['A', 'E', 'C']
A --> D: ['A', 'D']
A --> E: ['A', 'E']
A --> F: ['A', 'I', 'F']
A --> G: ['A', 'E', 'H', 'G']
A --> H: ['A', 'E', 'H']
A --> I: ['A', 'I']
A --> J: ['A', 'D', 'J']

For the MCGW graph
EEEE --> EEEE: ['EEEE']
EEEE --> WEWE: ['EEEE', 'WEWE']
EEEE --> EEWE: ['EEEE', 'WEWE', 'EEWE']
EEEE --> WWWE: ['EEEE', 'WEWE', 'EEWE', 'WWWE']
EEEE --> WEWW: ['EEEE', 'WEWE', 'EEWE', 'WEWW']
EEEE --> EWEE: ['EEEE', 'WEWE', 'EEWE', 'WWWE', 'EWEE']
EEEE --> EEEW: ['EEEE', 'WEWE', 'EEWE', 'WEWW', 'EEEW']
EEEE --> WWEW: ['EEEE', 'WEWE', 'EEWE', 'WWWE', 'EWEE', 'WWEW']
EEEE --> EWEW: ['EEEE', 'WEWE', 'EEWE', 'WWWE', 'EWEE', 'WWEW', 'EWEW']
EEEE --> WWWW: ['EEEE', 'WEWE', 'EEWE', 'WWWE', 'EWEE', 'WWEW', 'EWEW', 'WWWW']
```

Figure 1: Expected results for Q1.