

# Computational Thinking and Program Design

Assignment Nine (20 marks)  
(Amended on 1 May) (Due on 9 May 2022)

Rocky K. C. Chang

28 April 2022

## Instructions:

1. Submit your code for Q1, Q3 and Q4 and your answer to Q2 to the iLearning platform before 23:59 on 6 May.
  - For the code, submit `tryShortestPath1.py`, `tryShortestPath2.py`, `tryShortestPath3.py` and `graph.py`.
  - For your answer to Q2, submit a pdf file.
  - Compress all your files into a .7z file (using 7-zip). The file name should be "Your student ID".7z.
  - Submit the .7z file to the iLearning platform.
2. Late submission will not be accepted.
3. Observe also the penalty for plagiarism as stated in the Course Overview slides.

This set of questions is about the algorithm of finding a shortest path implemented in `findShortestPath()`. You may find `tryShortestPath0.py` which contains `findShortestPath()` and `printPath()` and a graph for our MCGW problem, and `graph.py` with this assignment. Answer the four questions below. **No global variables are allowed in your answers to Q1, Q3 and Q4.**

## Question 1

[5 MARKS] Write as much comment as possible on the code in `findShortestPath()`. We started doing that in our last online class. Please name this file as `tryShortestPath1.py`.

## Question 2

[5 MARKS] You may have already noticed that the code in `findShortestPath()` checks whether the start node is in the graph. However, it does not check for the end node. If you set the end node to an illegal state, such as "WEEW", would invoking `findShortestPath()` give you an error? Explain your answer.

**Solution:** No. According to the code in `findShortestPath()`, it will search for any given destination. If it is not found, the function will return `None`. However, the `printPath()` will give an error, because it assumes a path is given.

### Question 3

[5 MARKS] Now add two print statements to `findShortestPath()` in `tryShortestPath1.py`:

1. Add `"print('Call itself:',node,end,path)"` immediately before `newpath = ...` and
2. Add `"print('New path :', node,end,newpath)"` immediately after `newpath = ....`

These two print statements will help you understand how this program works by tracing the recursive functions calls and the variables' values. Run it and study the output. By counting the number of "Call itself" lines, you will find that there are a total number of 17 function calls, including the first time of calling `findShortestPath()`.

In this question, you are asked to modify `findShortestPath()` in `tryShortestPath1.py`, so that `findShortestPath()` will return the total number of function calls. That is, the modified function will also return both a shortest path and the total number of function calls. You cannot introduce additional parameters to the function. Please name this file as `tryShortestPath2.py`.

Please add the following testing code to `tryShortestPath2.py`. The expected results are given in Figure 1.

```
path, totalCall = findShortestPath(G,"EEEE","WWW")
printPath(path)
print()
print("The total number of recursive function calls is:", totalCall)
```

  

```
1. The man takes the goat from the east to the west.
2. The man takes only himself from the west to the east.
3. The man takes the cabbage from the east to the west.
4. The man takes the goat from the west to the east.
5. The man takes the wolf from the east to the west.
6. The man takes only himself from the west to the east.
7. The man takes the goat from the east to the west.
```

  

```
The total number of recursive function calls is: 17
```

Figure 1: The expected results for Q3.

### Question 4

[5 MARKS] In this last question, you are asked to modify `findShortestPath()` in `tryShortestPath1.py`, so that `findShortestPath()` will print out the sequence of function calls with the start node and end node. Please name this file as `tryShortestPath3.py`.

Please add the following testing code to `tryShortestPath3.py`. The expected results are given in Figure 2.

```
print("The sequence of recursive function calls:")
path, nextIndex = findShortestPath(G,"EEEE","WWW")
print()
print("The total number of recursive function calls is:", nextIndex-1)
print()
printPath(path)
```

The sequence of recursive function calls:

```
1: EEEE --> WWWW
2: WEWE --> WWWW
3: EEWE --> WWWW
4: WWWE --> WWWW
5: EWEE --> WWWW
6: WWEW --> WWWW
7: EWEW --> WWWW
8: WWWW --> WWWW
9: EEEW --> WWWW
10: WEWW --> WWWW
11: WEWW --> WWWW
12: EEEW --> WWWW
13: WWEW --> WWWW
14: EWEE --> WWWW
15: WWWE --> WWWW
16: EWEW --> WWWW
17: WWWW --> WWWW
```

The total number of recursive function calls is: 17

1. The man takes the goat from the east to the west.
2. The man takes only himself from the west to the east.
3. The man takes the cabbage from the east to the west.
4. The man takes the goat from the west to the east.
5. The man takes the wolf from the east to the west.
6. The man takes only himself from the west to the east.
7. The man takes the goat from the east to the west.

Figure 2: The expected results for Q4.