Computational Thinking and Program Design

Final Examination

Rocky K. C. Chang

22 June 2022

Name: _____ Student ID: _____ Total marks: _____

Please read the instructions below very carefully:

1. This examination will start at 13:10 and end at 16:10.
2. This is a take-home examination.
3. The exam paper and all the accompanied files are in iLearning.
4. There are 4 compulsory questions. Read the instructions in each question carefully.
5. There are 4 respective submission boxes in iLearning.
6. No late submissions will be accepted.
7. You can use any materials to help you answer the questions.
8. You cannot communicate and share information with anyone else in the world. A student who violates this either by giving out information to another student or receiving information from another student will fail in this course.
9. You cannot share the exam paper with others during the exam duration. Same consequence as the last point if violated.
10. You are required to stay in the Teams session until you finish the exam.

In this examination you will apply computational thinking and program design to solve another river-crossing riddle as described below.

> Albert is going to take his friends—Billy, Catherine, David, Eliza and Frank—to go over to the other side of a river in his boat in a minimum number of trips. There are several constraints below:
>
> 1. Only Albert can operate the boat.
> 2. Besides Albert, he can take at most two other people with him.
> 3. For whatever reason, Billy and Catherine cannot stay together without Albert.
> 4. For whatever reason, Catherine and David cannot stay together without Albert.
> 5. For whatever reason, David and Eliza cannot stay together without Albert.
> 6. For whatever reason, Eliza and Frank cannot stay together without Albert.

In your solution, you could assume they are initially on the east side of the river and their destination is the west side. Note that this is a natural extension to the MCGW problem. Similar to the MCGW problem, you are going to define the state of the problem as a 6-character string which describes the positions of Albert, Billy, Catherine, David, Eliza and Frank. Each character is either "E" (for east) or "W" (for west). The first character is the position of Albert, second Billy and so on. Please answer the four questions below for this riddle.

Question 1 (Solving the new river-crossing riddle)

[70 MARKS] You may find `newRiverRiddle.py`, `graph.py` and `myqueue.py` attached with this paper. You will implement the five functions in `newRiverRiddle.py`: `genStates()` (10 marks), `genGraph()` (10 marks), `isAStateLegal()` (10 marks), `isNeighbor()` (20 marks), and `printPath()` (20 marks). Read the function signatures carefully and the requirements below.

1. The graph will be implemented by a Python dictionary in which the key is a node, and the value is a list of the node's neighboring nodes.
2. You cannot modify `graph.py`, `myqueue.py` and the code in `newRiverRiddle.py`.
3. The printing format is similar to the MCGW problem, e.g.,
   - `Albert takes himself from the west to the east.`
   - `Albert takes David from the east to the west.`
   - `Albert takes Cathy and Eliza from the east to the west.`
4. Besides `myqueue.py` and `graph.py`, you cannot import other `.py` files.
5. If you cannot obtain the graph, you can uncomment the dictionary for the graph in `newRiverRiddle.py` and comment "`G = genGraph(listAllStates)`." If you uncomment the dictionary, you may lose the marks for `genStates()`, `genGraph()`, `isAStateLegal()`, and `isNeighbor()`.
6. Put `newRiverRiddle.py`, `graph.py` and `myqueue.py` into a folder called "`Your student ID`"`_Q1`. Compress it into "`Your student ID`"`_Q1.7z` and submit it to the iLearning submission box for Q1.

Question 2 (Time efficiency of `bfsTree()`)

[5 MARKS] In this question we are interested in the amount of work required by `bfsTree()` in `newRiverRiddle.py` to form a BFS tree for the new river-crossing riddle. A `Q2.py` is already prepared for you. You will write additional code to produce the outputs below, where `XXX` is a positive integer.

```
The number of enqueue() functions calls: XXX
The number of if-condition checking: XXX
```

Requirements for the solution and submissions.

- Since you are importing functions from `newRiverRiddle.py`, you have to put it with `Q2.py`.
- You cannot modify anything above the line "`# Write your code below.`"
- You cannot import other `.py` files and other functions in `newRiverRiddle.py`.
- You cannot modify `bfsTree()` to get your answers.
- You may use the functions in `graph.py` in your solution.
- If you cannot obtain the graph in Q1, you can use the dictionary for the graph in `newRiverRiddle.py` and comment "`G = genGraph(listAllStates)`."
- Put `newRiverRiddle.py`, `graph.py`, and `Q2.py` into a folder called "`Your student ID`"`_Q2`. Compress it into "`Your student ID`"`_Q2.7z` and submit it to the iLearning submission box for Q2.

Question 3, Part 1 (Adding another constraint to the riddle)

[5 MARKS] Now we impose an additional constraint on the riddle: Each female friend (i.e., Catherine and Eliza) cannot stay alone on either side of the river. There are two tasks in this part. First, you will find the set of nodes to be removed from the graph due to the new constraint. You will call this set `setNodeRemoved`. Second, you will eliminate the nodes in `setNodeRemoved` from the graph you obtain from Q1. You will write your code below the line "`# Write your code below.`" Requirements for the solution and submissions below.

- Since you are importing functions from `newRiverRiddle.py`, you have to put it with `Q3.py`.
- You cannot modify anything above the line "`# Write your code below.`" and below "`# (DO NOT REMOVE) Code for Rocky's testing.`"
- You cannot import other `.py` files and other functions in `newRiverRiddle.py`.
- You cannot generate a new graph using `genGraph()`.
- You may use the functions in `graph.py` in your solution.
- If you cannot obtain the graph in Q1, you can use the dictionary for the graph in `newRiverRiddle.py` and comment "`G = genGraph(listAllStates)`."
- See Part 2 for the submission instruction.

Question 3, Part 2 (Adding another constraint to the riddle)

[10 MARKS] Draw the backward BSF tree generated from the testing code in Part 1. Submit the drawing in a pdf file named `Q3-drawing.pdf`. Put `newRiverRiddle.py`, `graph.py`, `myqueue.py`, `Q3.py` and `Q3-drawing.pdf` into a folder called "`Your student ID`"`_Q3`. Compress it into "`Your student ID`"`_Q3.7z` and submit it to the iLearning submission box for Q3.

Question 4 (Building another BFS "tree" for finding all shortest paths)

[10 MARKS] Consider a 10-node graph in Figure 1 and the backward BFS tree rooted at A obtained from `bfsTree()` in `newRiverRiddle.py` in Figure 2.

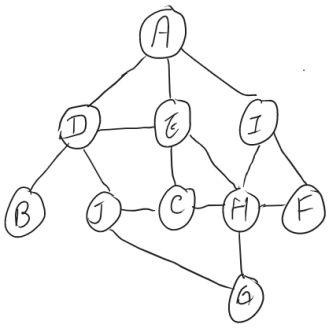

Figure 1: A 10-node graph for Q4.

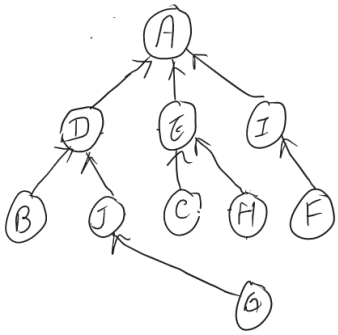

Figure 2: A backward BFS tree rooted at A for the graph in Figure 1.

By inspecting the graph, we know that there are three shortest paths for A to reach G: $A \to D \to J \to G$, $A \to E \to H \to G$, and $A \to I \to H \to G$. However, the backward BFS tree gives only one of them. One way to obtaining all three shortest paths is to add additional links to Figure 2 to cover all three paths. Figure 3 shows such a modified "tree" which is actually not a tree. Instead of having only a single parent node, each node now has a set of parent nodes, e.g., $\{E, I\}$ for H and $\{H, J\}$ for G. Your task is to modify `bfsTree()` in `newRiverRiddle.py` to return the modified tree. The new function is called `modifiedBsfTree()`.

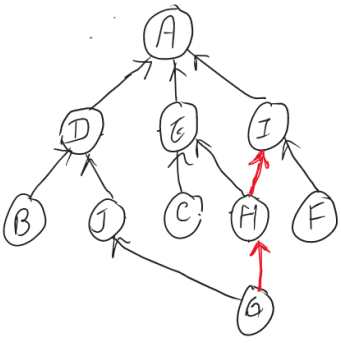

Figure 3: Adding additional links to Figure 2 for covering all three shortest paths.

`Q4.py` is prepared for you. Your function will be tested using the original graph (i.e., the one produced in Q1). You will write your code below the line "`# Write your code below.`" Requirements for the solution and submissions below.

- Since you are importing functions from `newRiverRiddle.py`, you have to put it with `Q4.py`.
- You cannot modify anything above the line "`# Write your code below.`" and below "`# (DO NOT REMOVE) Code for Rocky's testing.`"
- You cannot import other `.py` files and other functions in `newRiverRiddle.py`.
- The values in the dictionary for the modified tree must be a set of parent nodes.
- If you cannot obtain the graph in Q1, you can use the dictionary for the graph in `newRiverRiddle.py` and comment "`G = genGraph(listAllStates)`."
- Put `newRiverRiddle.py`, `graph.py`, `myqueue.py`, and `Q4.py` into a folder called "`Your student ID`"`_Q4`. Compress it into "`Your student ID`"`_Q4.7z` and submit it to the iLearning submission box for Q4.

———— END OF THIS EXAM PAPER ————