EC504 Advanced Data Structures & Algorithms Project Report 5/1/22

i. Team information

Name: Olivier Chen BU ID: U33604671 SCC user name: ochen

ii. Abstract

Created a web-based application for visualizing certain algorithms. These include depth-first search, breath-first search, and Dijkstra's. The server is run through Flask and the algorithms are written in Python in the Algorithms.py file. Because this is a visualization project, Cytoscape.js is used as a JavaScript library to display the graph. Certain functions like adding edges, vertices, and running the algorithm are in the cytoscape.js file. Finally, the front end is built with HTML and CSS in the algos.html and stylesheet.css file within the src folder. The cytoscape library is within the node modules folder.

iii. Instructions

Simply download the FinalProject folder from my ochen directory. Open the folder inside a code editor (I used VSCode). Go to the server.py file and click run and then open the link on your local browser.

```
PROBLEMS TERMINAL DEBUG CONSOLE

(base) olivierchen@Oliviers-MacBook-Pro FinalProject % /Users/olivierchen/opt/anaconda3/bin/python /Users/olivierchen/Downloads/FinalProject/src/server.py

* Serving Flask app "server" (lazy loading)

* Environment: production

WARNING: This is a development server. Do not use it in a production deployment.

Use a production WSGI server instead.

* Debug mode: on

* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

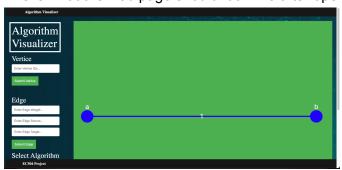
* Restarting with watchdog (fsevents)

* Debugger is active!

* Debugger PIN: 134-484-575
```

Make sure to have Flask installed.

This is what the web page should look like after opening the http://

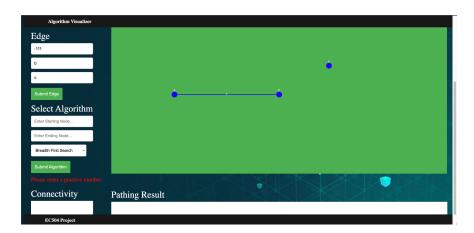


Here you can add vertices by typing in your desired vertice name. Note that you may have to zoom out on the graph visualization to see the new vertice created.

You can also add weighted edges between any two vertices.

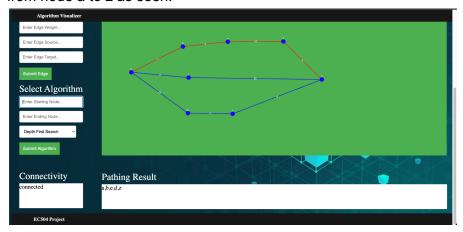
Lastly, if you scroll down, you can run DFS, BFS, or Dijkstra's on a start and ending node. After running, the path should glow red on the visualization and the list of all vertices in the path should appear below.

There is error checking for duplicate vertices, edges, incorrect edge or negative edge weights, and checking if vertices exist. DFS and BFS ignores edge weights and Dijkstra's uses edge weights.

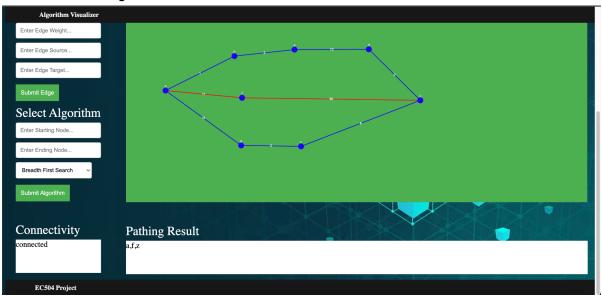


iv. Sample results

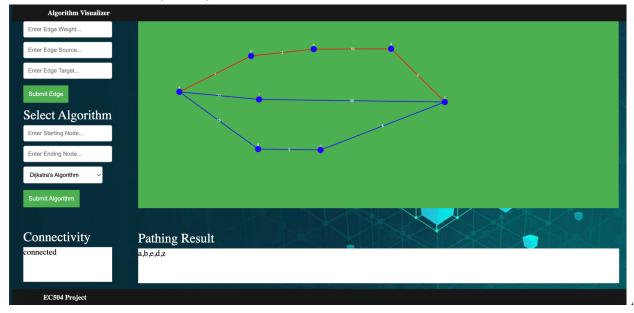
Results for DFS: running DFS: the algorithm will go down the first path depth-wise and reach from node a to z as seen.



Results for BFS: For BFS, the algorithm will take the path that contains the least amount of nodes to reach target z.



Results for Dijkstra's: In Dikkstra's algorithm, the weights matter so the algorithm will take the path with least total weights regardless of number of nodes.



It may be hard to see from the screenshot but the path shown here has the most nodes but the least total cost of 15 versus the cost of 61 and 26 for the paths below with less number of nodes.

v. References

Using the Cytoscape.js library for building the graph visualization. https://js.cytoscape.org/