

## Team Information

Team information should list members with names, BU ID and SCC usernames.

Iker Zamora - BU ID: U83267103 - SCC username: zamorai

Ivan Isakov - BU ID: U52060548 - SCC username: isakovi

Ryan Sullivan - BU ID: U73687595 - SCC username: ryanbs

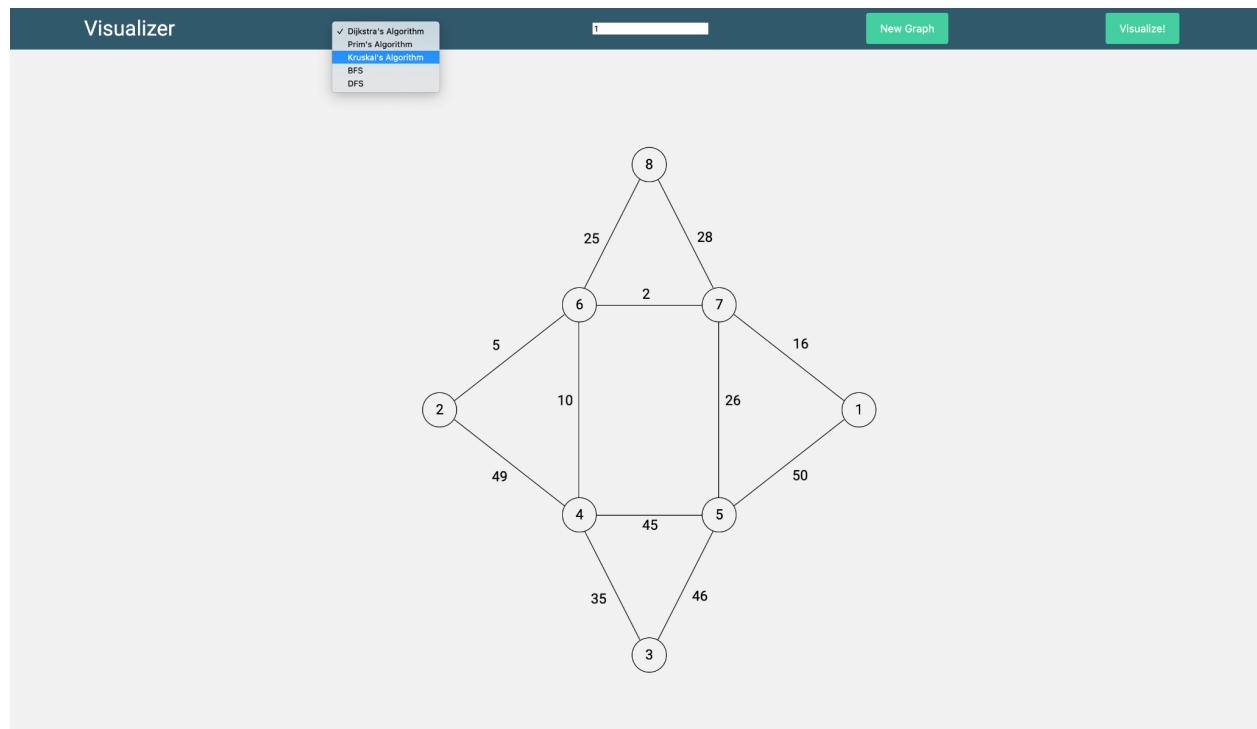
## Abstract

This project aims to visualise some of the graph traversal algorithms that we have studied, such as: BFS, DFS, Prim-Jarnik, Kruskal and Dijkstra. We decided to use React.js and CSS for this implementation since it has all the necessary features for visualisation. The project itself, we believe, gives an example of how these algorithms work, so that it is easier to understand how the algorithms explore the graphs and converge to a solution.

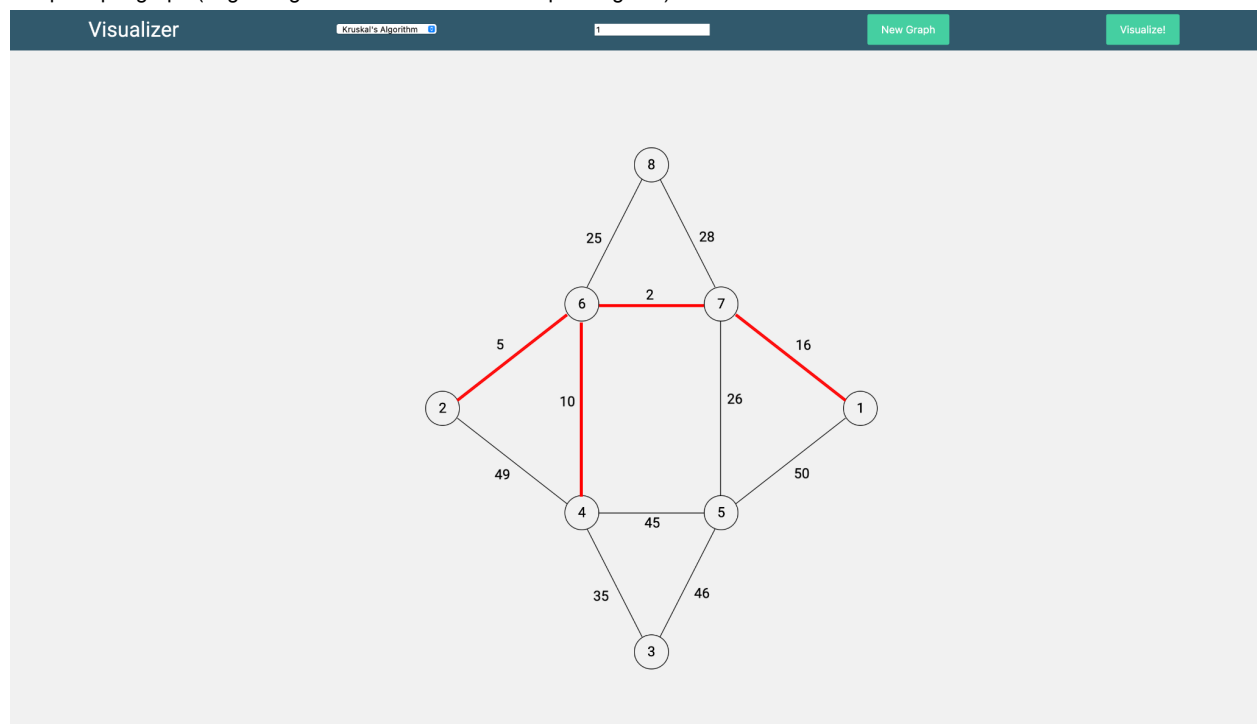
## Instructions for running

For the TA's convenience, we have hosted all the project files in our github repository (<https://github.com/zamorai/visualizer>), and also hosted the files through netlify, so all they need to do is follow this link to reach our project <https://nifty-perlman-7d908e.netlify.app/>. In order to run the project, simply select the algorithm to run on the top left corner, and input a number between 1 - 8 in the start node (or end node when applicable) and simply press visualize. If you want to see different graphs to corroborate the algorithms work, simply press the new graph button. If you want to see all of the source code, follow the github repo and search the files. The primary bulk of the project, where all the algorithms are hosted, is in the file App.js, and the visual aspects are in the styles.css files, Graph.js and Header.js files. In case the website is not functioning properly, please download the files using the git bash terminal, and install nodejs in your computer. Run **git clone** <https://github.com/zamorai/visualizer.git>, followed by switching into the directory you cloned the repo in. Then run **npm install** to install all the dependencies for the project. Finally run **npm start** to start the project, it will automatically start a server in port 3000 on your browser and you will be able to see the same project as the one hosted on the website.

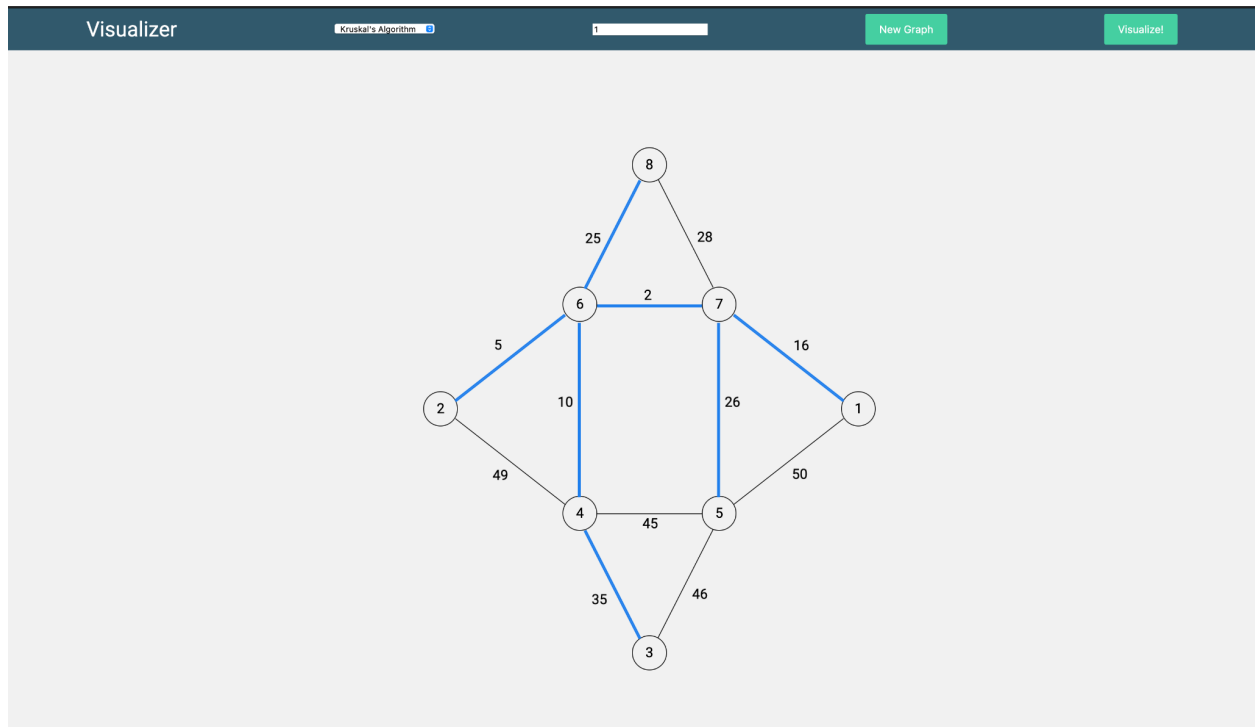
## Sample results



Sample input graph (edge weights are next to their corresponding line)



Intermediary graph (Kruskal's)



Solved graph (Kruskal's)

The previous images show the input graph in all of its stages: initial, working, and completed. As the graph is being solved the currently analyzed edges/vertex are highlighted in red. Once the graph has been solved, either by calculating the MST or finding a path from one initial vertex to the target vertex, the path or tree will be highlighted in blue.

## References

v. References include a list of sources your team used in developing the software. They can include web sites, or reference papers.

<https://www.programiz.com/dsa/prim-algorithm>

<https://www.programiz.com/dsa/dijkstra-algorithm>

<https://stackoverflow.com/questions/41043419/reactjs-onclick-state-change-one-step-behind>