**Report 1**

Lab 1; Exploration of path optimization

104349 Grafos, Topología y Geometría Discreta

Thijs Rood, 1644492

17/02/2022

In this lab I tested two algorithms to search to the network of a graph for separate components. The algorithms used are breadth first search (BFS) & depth first search (DFS) respectively. The dataset is a network from social media platform LastFM. The questions will be answered in this report.

*Report Questions I*

1. I made two small test networks to quickly check the workings of my functions so I wouldn’t have to run the large csv file from LastFM. One of the small networks contained one larger component and the other network contained multiple disconnected components. I also tested the correctness with an additional algorithm found in the documentation.
2. In the LastFm dataset there is only 1 component. Meaning everyone is connected.
3. When the target items are closer to the source, in a more tree-like or breadthened structure, BFS is logically faster. When targets are further from a source in a more linear structure, DFS is faster. This is because both algorithms check the nodes in a horizontal and vertical path respectively.
4. For the particular case of the LastFM dataset the BFS is faster. This might be explained by the large amount of connections some of the social network’s users have. This can be determined by putting a time object in the python code, it can measure the time needed to perform one of the algorithms. In my case the DFS sometimes even overloaded my kernel.

*Report Questions II*

1. I used the breadth first algorithm in the how\_many\_degrees function as it would be easy to add up the distance for each node that is passed. In the end all distances are computed and can be filtered so only the smallest distances are accounted for.
2. Done.
3. As the how\_many\_degrees function had already been implemented it was straightforward to make a function that would compute the largest distance between two undetermined nodes in a network. First an iteration over each possible node combination is done, for each of these the how\_many\_degrees function had been applied. Finally the largest of these values had been returned.