PEER REVIEW REPORT: NICLAS LINDMARK

By Erik Brechter

SECTION 1: CORE ASSIGNMENT

Q1: Does the application run? → YES

Q2: Does the application display the complete map of tram lines? → YES

Q3: Is it possible to query shortest path between any two points? → YES

SECTION 2: OPTIONAL TASKS

B1: Is the submission successfully accounting for Bonus Part 1? → NO

B2: Is the submission successfully accounting for Bonus Part 2 → YES

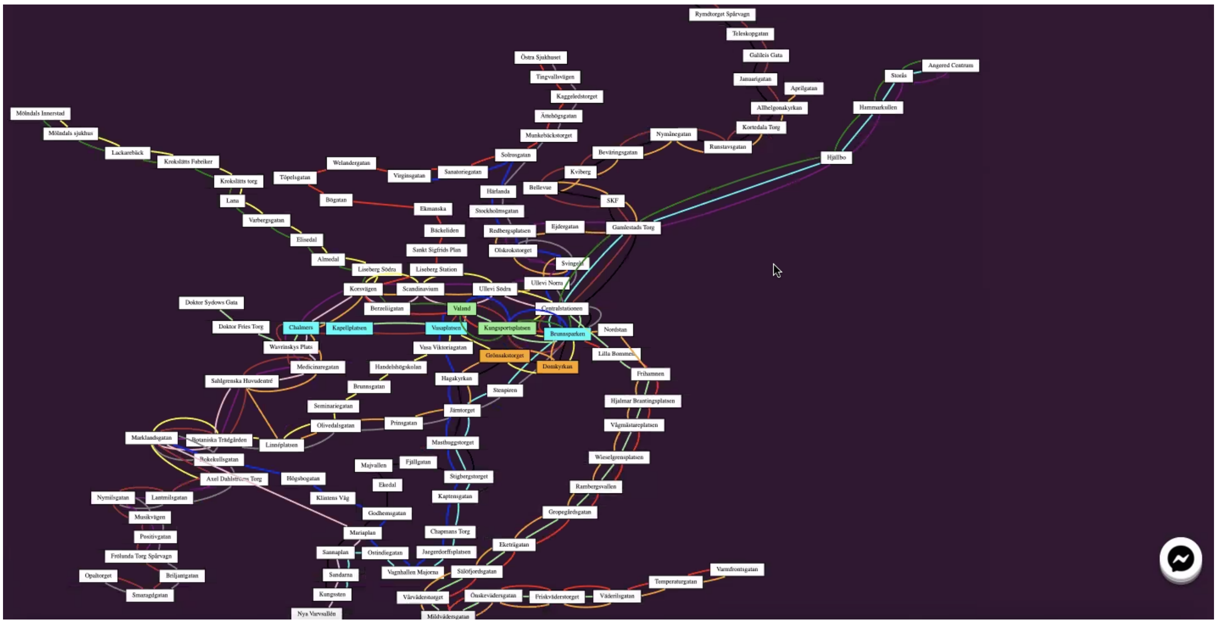
SECTION 3: CODE QUALITY

Djikstra has been implemented as intended, there is just one definition of the function itself and different distanced are obtained by changing the cost function.

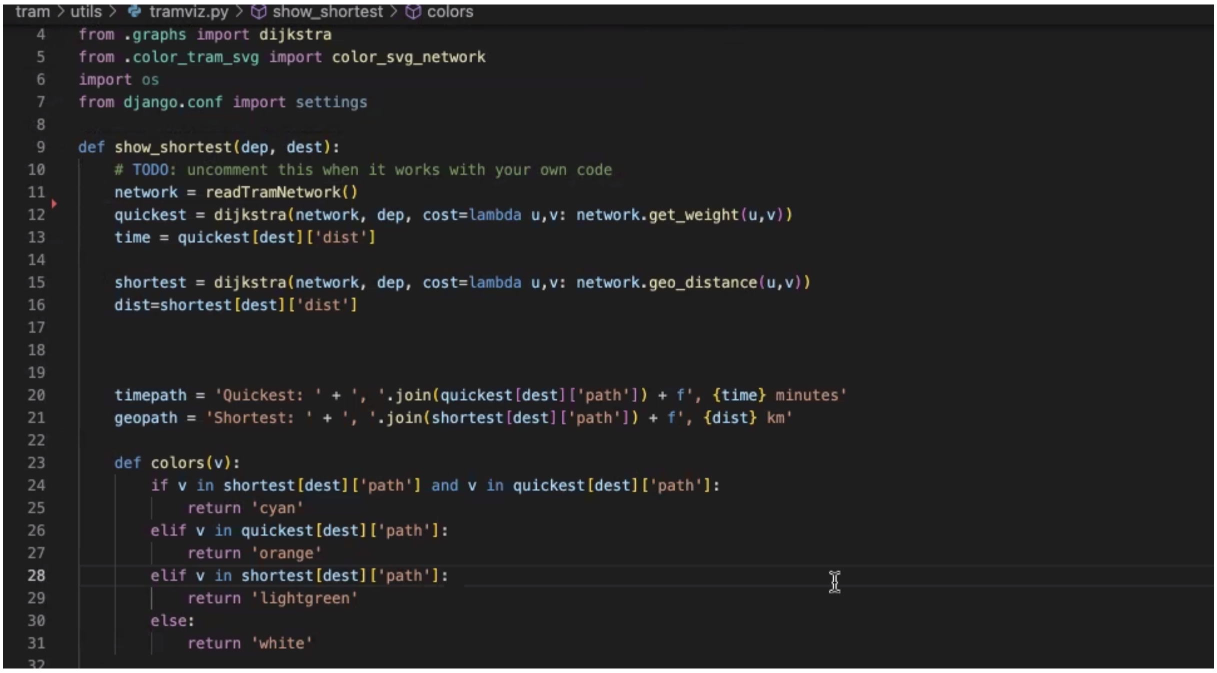
The code from lab 2 has been copied over and changed where it was necessary. Overall the code is very efficient.

SECTION 4: SCREENSHOTS:

Screenshot 1: Correctly displays the shortest path and quickest path from Chalmers to Brunnsparken.



Screenshot 2: Screenshot of show_shortest() function.



```python
from .graphs import dijkstra
from .color_tram_svg import color_svg_network
import os
from django.conf import settings

def show_shortest(dep, dest):
    # TODO: uncomment this when it works with your own code
    network = readTramNetwork()
    quickest = dijkstra(network, dep, cost=lambda u,v: network.get_weight(u,v))
    time = quickest[dest]['dist']

    shortest = dijkstra(network, dep, cost=lambda u,v: network.geo_distance(u,v))
    dist=shortest[dest]['dist']



    timepath = 'Quickest: ' + ', '.join(quickest[dest]['path']) + f', {time} minutes'
    geopath = 'Shortest: ' + ', '.join(shortest[dest]['path']) + f', {dist} km'

    def colors(v):
        if v in shortest[dest]['path'] and v in quickest[dest]['path']:
            return 'cyan'
        elif v in quickest[dest]['path']:
            return 'orange'
        elif v in shortest[dest]['path']:
            return 'lightgreen'
        else:
            return 'white'
```