

Peer Review of Filip Sjölander's project

Josef Jakobson

March 2, 2023

1 Core Assignment

1.1 Does the application run?

Yes

1.2 Does the application display the complete map of tram lines?

Yes

1.3 Is it possible to query shortest path between any two points?

Yes

2 Optional Tasks

No bonus tasks were implemented

3 Code Quality

The code is in general very well structured and adheres to standards of object oriented programming where needed. On the topic of boilerplate code, the only code that appears more than once is `graphs.py` and `trams.py` as per the assignment, other than that there is no code duplication regarding the implementation of `tramviz.py`. Dijkstras is only defined once, and when getting the quickest and shortest paths, `dijkstras` is called simply with two different cost functions as arguments.

4 Screenshots

Chalmers-Mildvädersgatan

Quickest: Chalmers, Kapellplatsen, Vasaplatsen, Grönsaktorget, Domkyrkan, Brunnsparken, Nordstan, Frihamnen, Hjalmar Brantingsplatsen, Vågmästareplatsen, Wieselgrensplatsen, Rambergsvallen, Gropegårdsgatan, Eketräsgatan, Sällöfordsgatan, Våräderstorget, Mildvädersgatan - 20 minutes
Shortest: Chalmers, Kapellplatsen, Vasaplatsen, Våland, Kungssportsplatsen, Brunnsparken, Lilla Bommen, Frihamnen, Hjalmar Brantingsplatsen, Vågmästareplatsen, Wieselgrensplatsen, Rambergsvallen, Gropegårdsgatan, Eketräsgatan, Sällöfordsgatan, Våräderstorget, Mildvädersgatan - 8.2 km

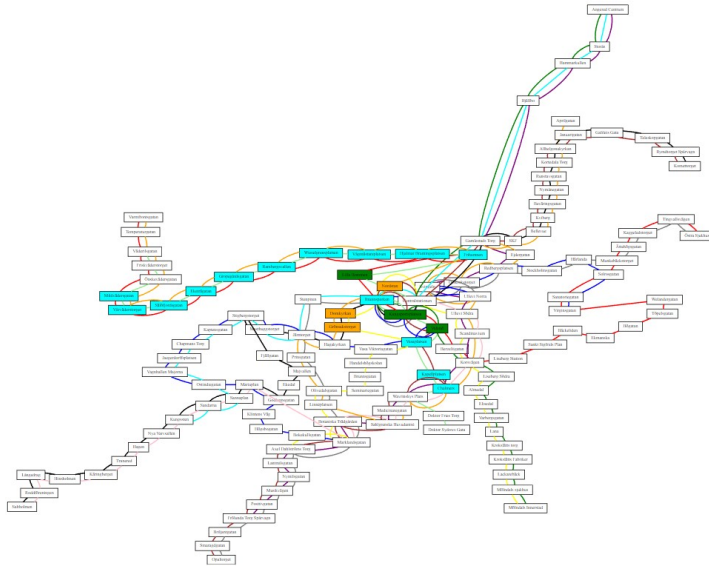


Figure 1: Shortest path visualized.

```
8
9 def show_shortest(dep, dest):
10     network = readTramNetwork()
11
12     quickest = dijkstra(network, dep, cost=lambda u,v: network.get_weight(u,v))[dest]
13
14     shortest = dijkstra(network, dep, cost=lambda u,v: network.geo_distance(u,v))[dest]
15
16     timepath = 'Quickest: ' + ', '.join(quickest["path"]) + ' - {} minutes'.format(round(quickest["distance"], 1))
17     geopath = 'Shortest: ' + ', '.join(shortest["path"]) + ' - {} km'.format(round(shortest["distance"], 1))
18
19     def colors(v):
20         if v in quickest["path"] and v in shortest["path"]:
21             return 'cyan'
22         elif v in shortest["path"]:
23             return 'green'
24         elif v in quickest["path"]:
25             return 'orange'
26         else:
27             return 'white'
28
29
30     # this part should be left as it is:
31     # change the SVG image with your shortest path colors
32     color_svg_network(colormap=colors)
33     # return the path texts to be shown in the web page
34     return timepath, geopath
35
```

Figure 2: Code for `showshortest()`.