

DAT515-DIT515 /
lab-3-a-web-application-for-tram-network-m

<> Code

Issues

Pull requests 1

Actions

Projects

Security

Insights

lab-3-a-web-application-for-tram-network-m / lab3 / tram / utils / tramviz.py

...

andyvungoc Lab3

15 hours ago

43 lines (36 loc) · 1.83 KB

Code

Blame

Raw



```
1 from .trams import readTramNetwork #, specialized_transition_time, specialized_geo_distance, sp
2 from .graphs import dijkstra
3 from .color_tram_svg import color_svg_network
4 import os
5 from django.conf import settings
6
7
8 ✓ def show_shortest(dep, dest):
9     # TODO: uncomment this when it works with your own code
10    network = readTramNetwork()
11    # TODO: replace this mock-up with actual computation using dijkstra.
12    # First you need to calculate the shortest and quickest paths, by using appropriate
13    # cost functions in dijkstra().
14    # Then you just need to use the lists of stops returned by dijkstra()
15    # If you do Bonus 1, you could also tell which tram lines you use and where changes
16    # happen. But since this was not mentioned in lab3.md, it is not compulsory.
17
18    quickest = dijkstra(network, dep, cost=lambda u,v: network.transition_time(u,v))[dest]
19    print("Quickest"+ str(quickest))
20    shortest = dijkstra(network, dep, cost=lambda u,v: network.geo_distance(u,v))[dest]
21    print("Shortest"+str(shortest))
22
23    timepath = f'Quickest: {" - ".join(quickest["path"])}', {quickest["weight"]} minutes'
24    geopath = f'Shortest: {" - ".join(shortest["path"])}', {round(shortest["weight"], 2)} km'
25    #print(specialize_stops_to_lines(network))
26    #print(specialized_transition_time(specialize_stops_to_lines(network)))
27 ✓ def colors(v):
28     if (v in shortest['path']) and (v in quickest['path']):
29         return 'cyan'
30     elif v in shortest['path']:
31         return 'lightgreen'
32     elif v in quickest['path']:
33         return 'orange'
34     else:
35         return 'white'
36
37     # this part should be left as it is:
38     # change the SVG image with your shortest path colors
39     color_svg_network(colormap=colors)
40     # return the path texts to be shown in the web page
41     return timepath, geopath
```