

39 lines (28 loc) · 1.35 KB

Code

Blame

Raw



```
1  # visualization of shortest path in Lab 3, modified to work with Django
2
3  from .trams import readTramNetwork
4  from .graphs import dijkstra
5  from .color_tram_svg import color_svg_network
6  import os
7  from django.conf import settings
8
9  ✓ def show_shortest(dep, dest):
10     network = readTramNetwork()
11
12     shortest_timepath = dijkstra(network, dep, cost=lambda u,v: network.get_weight(u,
13     shortest_geopath = dijkstra(network, dep, cost=lambda u,v: network.geo_distance(u,
14
15     quickest = shortest_timepath[dest]["path"]
16     shortest = shortest_geopath[dest]["path"]
17
18
19     timepath = 'Quickest: ' + ', '.join(quickest) + ", " + str(shortest_timepath[dest]
20     geopath = 'Shortest: ' + ', '.join(shortest) + ", " + str(round(shortest_geopath[d
21
22
23  ✓ def colors(v):
24     if v in shortest and v in quickest:
25         return "cyan"
26     elif v in shortest and v not in quickest:
27         return 'green'
28     elif v in quickest and v not in shortest:
29         return "orange"
30     else:
31         return 'white'
32
33
34     # this part should be left as it is:
35     # change the SVG image with your shortest path colors
36     color_svg_network(colormap=colors)
37     # return the path texts to be shown in the web page
38     return timepath, geopath
```