project_notebook-zh

September 20, 2019

1 123

A*“Google-maps”

```
In [13]: # Run this cell first!

         from helpers import Map, load_map, show_map
         from helper import Maps, load_maps, show_maps
         from student_code import shortest_path

         %load_ext autoreload
         %autoreload 2
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

1.0.1 Map

```
In [2]: map_10 = load_map('map-10.pickle')
        show_map(map_10)
```

show_maps(map_10)Jupyter.html

1022

MapA *intersectionsroads

Intersections

intersections

10xy

```
In [3]: map_10.intersections
```

```
Out[3]: {0: [0.7798606835438107, 0.6922727646627362],
         1: [0.7647837074641568, 0.3252670836724646],
         2: [0.7155217893995438, 0.20026498027300055],
         3: [0.7076566826610747, 0.3278339270610988],
         4: [0.8325506249953353, 0.02310946309985762],
         5: [0.49016747075266875, 0.5464878695400415],
 6: [0.8820353070895344, 0.6791919587749445],
 7: [0.46247219371675075, 0.6258061621642713],
 8: [0.11622158839385677, 0.11236327488812581],
 9: [0.1285377678230034, 0.3285840695698353]}
```

Roads

roadsi roads[i]i

```
In [4]: # this shows that intersection 0 connects to intersections 7, 6, and 5
        map_10.roads[0]

Out[4]: [7, 6, 5]

In [5]: # This shows the full connectivity of the map
        map_10.roads

Out[5]: [[7, 6, 5],
         [4, 3, 2],
         [4, 3, 1],
         [5, 4, 1, 2],
         [1, 2, 3],
         [7, 0, 3],
         [0],
         [0, 5],
         [9],
         [8]]

In [7]: # map_40 is a bigger map than map_10
        map_40 = load_map('map-40.pickle')
        show_map(map_40)
```

show_maps(map_40)Jupyter.html

1.0.2

40039

show_map

- start - “”
- goal - “”
- path -

```
In [8]: # run this code, note the effect of including the optional
        # parameters in the function call.
        show_map(map_40, start=5, goal=34, path=[5,16,37,12,34])
```

1.0.3

```
student_code.pyFile > Open
   show_mappath[5, 16, 37, 12, 34]
```

```
In [9]: %%bash
        > shortest_path(map_40, 5, 34)
        [5, 16, 37, 12, 34]
```

```
bash: line 1: syntax error near unexpected token `('
bash: line 1: `> shortest_path(map_40, 5, 34)'
```

```
In [14]: import time
         start = time.clock()

         path = shortest_path(map_40, 5, 34)
         if path == [5, 16, 37, 12, 34]:
             print("great! Your code works for these inputs!")
         else:
             print("something is off, your code produced the following:")
             print(path)

         elapsed = (time.clock() - start)
         print("Time used:",elapsed)
```

```
shortest path called
great! Your code works for these inputs!
Time used: 0.010691000000000006
```

1.0.4

1.
2. A*
3.
4.

  ″″

```
In [15]: from test import test

         start = time.clock()

         test(shortest_path)

         elapsed = (time.clock() - start)
         print("Time used:",elapsed)
```

```
shortest path called
shortest path called
shortest path called
All tests pass! Congratulations!
Time used: 0.05096299999999987
```

```
In [ ]:
```