

Report

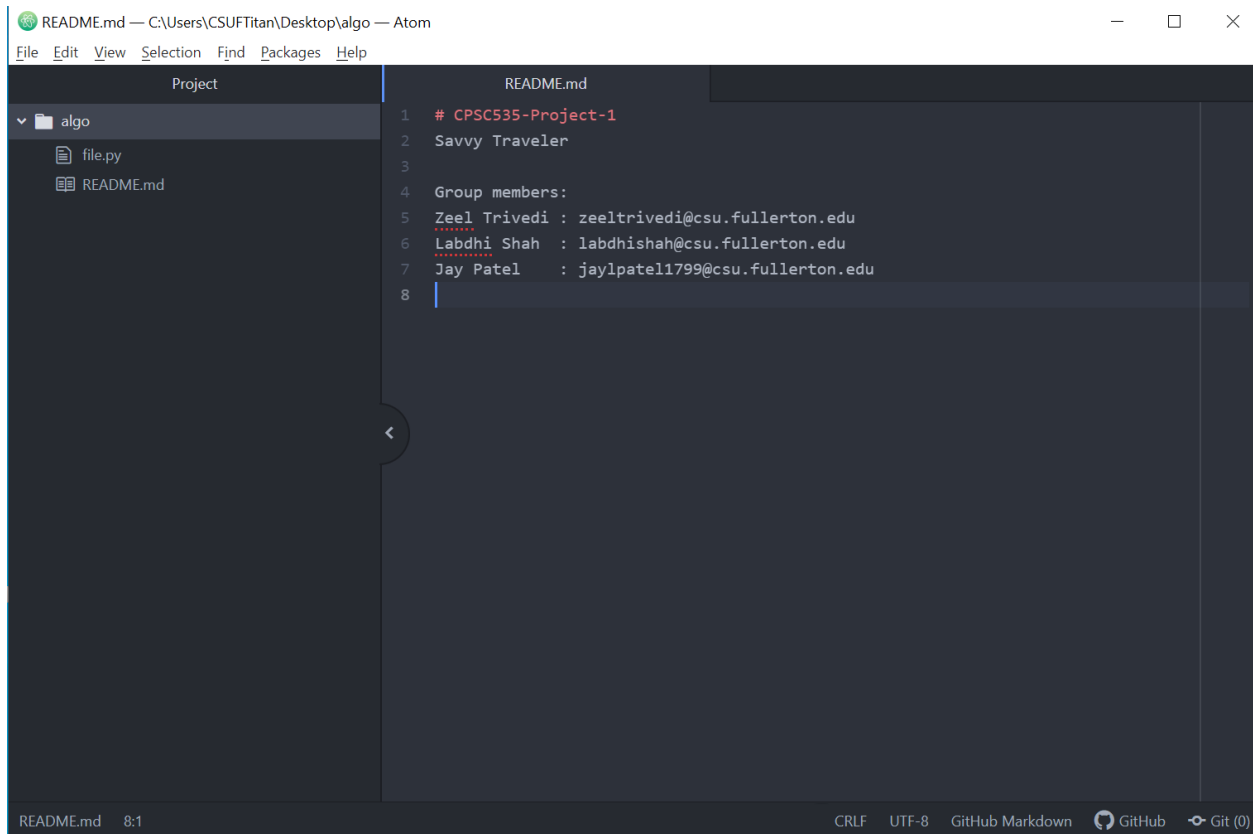
Project 1: Savvy Traveler

Group Members

Zeel Trivedi : zeeltrivedi@csu.fullerton.edu

Labdhi Shah : labdhishah@csu.fullerton.edu

Jay Patel : jaylpatel1799@csu.fullerton.edu



```
1 # CPSC535-Project-1
2 Savvy Traveler
3
4 Group members:
5 Zeel Trivedi : zeeltrivedi@csu.fullerton.edu
6 Labdhi Shah : labdhishah@csu.fullerton.edu
7 Jay Patel : jaylpatel1799@csu.fullerton.edu
8
```

Pseudocode description:

$Z = 8$ #number of nodes

I, j, k used in for loop to visit the each node

x, y source and destination node

$\text{MIN_PROBABILITY} = 0$ #to store the minimum probability of particular node

$\text{maps} = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']$ # nodes name

Pseudocode for the given problems:

project function to find the maximum distance between the source and destination node.

def project(graph, direction):

 solve = list(map(lambda i: list(map(lambda j: j, i)), graph))

 path = list(map(lambda i: list(map(lambda j: j, i)), direction))

 for k in range(Z):

 for i in range(Z):

 for j in range(Z):

 if solve[i][j] < solve[i][k] * solve[k][j]:

 path[i][j] = path[i][k] + path[k][j]:

to select the maximum distance between the two nodes via using the junction

 solve[i][j] = max(solve[i][j], solve[i][k] * solve[k][j])

to find the maximum probability of a city

 maxProb = 0 ## initializing the maximum probability to 0

 ans = '' ## to store the final answer and initialized it as NULL

 for i in range(Z):

 tmp = 1 ## tmp variable to store the probability of a city

```

    for j in range(Z):
        tmp *= solve[i][j]
    if tmp > maxProb:    ## compare tmp* with maxProb, and print the maximum
        maxProb = tmp    ## select that path which has the maximum value
        ans = i
    print("Probability:", solve[x][y])
    print("Path:", "->".join(list(path[x][y])))
    print("City:",maps[ans])
graph = []
direction = []
for i in range(Z):
    tmp = []
    tmp1 = []
    for j in range(Z):
        tmp.append(MIN_PROBABILITY)
        tmp1.append("")
    graph.append(tmp)
    direction.append(tmp1)

##explicitly describing the edges with values
##ed = [[0, 1], [0, 2], [0, 3], [1, 2], [1, 4], [1, 5], [2, 5], [3, 5], [3, 6], [4, 5], [4, 7], [5, 6], [5, 7],
[6, 7]]
##probs = [0.8, 0.7, 0.9, 0.8, 0.6, 0.6, 0.9, 0.6, 0.8, 0.8, 0.6, 0.7, 0.7, 0.9]
    direction[i[0]][i[1]] = maps[i[x]] + maps[i[y]]
    index += 1
project(graph,direction)

```

Brief Description on how to run the code:

- 1) In the given code user must change vertices and probability as per the graph.
- 2) There is three programs with different inputs for given graph with their probability as below:
Input 1: ed = [[0, 1], [0, 2], [0, 3], [1, 2], [1, 4], [1, 5], [2, 5], [3, 5], [3, 6], [4, 5], [4, 7], [5, 6], [5, 7], [6, 7]]
prob = [0.8, 0.7, 0.9, 0.8, 0.6, 0.6, 0.9, 0.6, 0.8, 0.8, 0.6, 0.7, 0.7, 0.9]
Input 2: ed= [[0,1], [1,2], [2,5],[5,4],[4,1], [4,3],[3,0],[3,6],[6,7],[7,4]]
probs= [0.8, 0.6 ,0.8, 0.9, 0.7, 0.8, 0.9, 0.8, 0.9,0.6]
Input 3: ed= [[0,1],[1,2],[2,3],[3,0],[0,4],[4,5],[5,6],[6,7],[7,4],[7,3],[6,2], [5,1]]
probs= [0.8,0.6,0.9,0.9,0.8,0.6,0.9,0.6,0.8,0.7,0.6,0.7]
- 3) After giving required inputs user should run the code.

Results:

Example 1

The screenshot shows a Windows Command Prompt window on the left and a Python IDE window on the right. The Command Prompt shows the user navigating to the Desktop, then to a directory named 'algo', and running 'python 1st.py'. The output shows the probability 0.63, the path A->C->F, and the city F. The Python IDE shows the code for '1st.py', which includes a README.md file and a script that defines a graph, sets probabilities, and runs a project function.

```

Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\CSUFTitan>cd Desktop
C:\Users\CSUFTitan\Desktop>cd algo
C:\Users\CSUFTitan\Desktop\algo>python 1st.py
Probability: 0.63
Path: A->C->F
City: F
C:\Users\CSUFTitan\Desktop\algo>

README.md
1st.py
40 direction = []
41 for i in range(Z):
42     tmp = []
43     tmp1 = []
44     for j in range(Z):
45         tmp.append(MIN_PROBABILITY)
46         tmp1.append("")
47     graph.append(tmp)
48     direction.append(tmp1)
49
50 #input for 1st example:
51 ed = [[0, 1], [0, 2], [0, 3], [1, 2], [1, 4], [1, 5], [2, 5], [3, 5], [3, 6], [4, 5],
52 probs = [0.8, 0.7, 0.9, 0.8, 0.6, 0.6, 0.9, 0.6, 0.8, 0.8, 0.6, 0.7, 0.7, 0.9]
53
54
55
56
57 index = 0
58 for i in ed:
59     graph[i[0]][i[1]] = probs[index]
60     graph[i[1]][i[0]] = probs[index]
61     direction[i[0]][i[1]] = maps[i[0]] + maps[i[1]]
62     direction[i[1]][i[0]] = maps[i[1]] + maps[i[0]]
63     index += 1
64
65 project(graph,direction)
66
CRLF UTF-8 Python GitHub Git (0)

```

Example 2:

```
Command Prompt
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\CSUFTitan>cd Desktop
C:\Users\CSUFTitan\Desktop>cd algo
C:\Users\CSUFTitan\Desktop\algo>python 1st.py
Probability: 0.63
Path: A->C->F
City: F

C:\Users\CSUFTitan\Desktop\algo>python 2nd.py
Probability: 0.5184000000000001
Path: C->F->E->D->A
City: D

C:\Users\CSUFTitan\Desktop\algo>

2nd.py
46 graph = []
47 direction = []
48 for i in range(Z):
49     tmp = []
50     tmp1 = []
51     for j in range(Z):
52         tmp.append(MIN_PROBABILITY)
53         tmp1.append("")
54     graph.append(tmp)
55     direction.append(tmp1)
56
57 #input for 2nd example:
58 ed= [[0,1], [1,2], [2,5],[5,4],[4,1], [4,3],[3,0],[3,6],[6,7],[7,4]]
59 probs= [0.8, 0.6, 0.8, 0.9, 0.7, 0.8, 0.9, 0.8, 0.9, 0.6]
60
61
62
63 index = 0
64 for i in ed:
65     graph[i[0]][i[1]] = probs[index]
66     graph[i[1]][i[0]] = probs[index]
67     direction[i[0]][i[1]] = maps[i[0]] + maps[i[1]]
68     direction[i[1]][i[0]] = maps[i[1]] + maps[i[0]]
69     index += 1
70
71 project(graph,direction)
72
```

Example 3

```
Command Prompt
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\CSUFTitan>cd Desktop
C:\Users\CSUFTitan\Desktop>cd algo
C:\Users\CSUFTitan\Desktop\algo>python 1st.py
Probability: 0.63
Path: A->C->F
City: F

C:\Users\CSUFTitan\Desktop\algo>python 2nd.py
Probability: 0.5184000000000001
Path: C->F->E->D->A
City: D

C:\Users\CSUFTitan\Desktop\algo>python 3rd.py
Probability: 0.6480000000000001
Path: E->A->D->C
City: A

C:\Users\CSUFTitan\Desktop\algo>

3rd.py
46 direction = []
47 for i in range(Z):
48     tmp = []
49     tmp1 = []
50     for j in range(Z):
51         tmp.append(MIN_PROBABILITY)
52         tmp1.append("")
53     graph.append(tmp)
54     direction.append(tmp1)
55
56 #input for 3rd example:
57 ed= [[0,1],[1,2],[2,3],[3,0],[0,4],[4,5],[5,6],[6,7],[7,4],[7,3],[6,2],[5,1]]
58 probs= [0.8,0.6,0.9,0.9,0.8,0.6,0.9,0.6,0.8,0.7,0.6,0.7]
59
60
61
62
63 index = 0
64 for i in ed:
65     graph[i[0]][i[1]] = probs[index]
66     graph[i[1]][i[0]] = probs[index]
67     direction[i[0]][i[1]] = maps[i[0]] + maps[i[1]]
68     direction[i[1]][i[0]] = maps[i[1]] + maps[i[0]]
69     index += 1
70
71 project(graph,direction)
72
```