Elements of AI - Programming Assignment 1 – Read Me

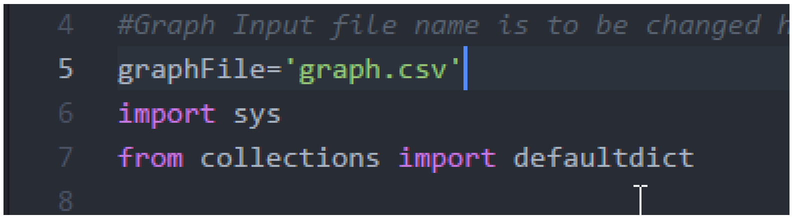
Ganesh Nagarajan (gnagaraj) & Siddharth Jayasankar (sidjayas)

Program:

The program is contained in the Python file “PA1_BFS_DFS_ID.py”

Map Input:

The Name of the CSV file needs to be specified at line 5 of the program.

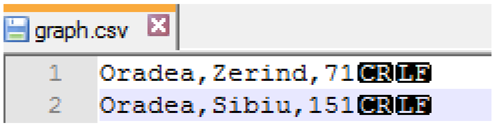


```
4  #Graph Input file name is to be changed h
5  graphFile='graph.csv'
6  import sys
7  from collections import defaultdict
8
```

The Map input for this program should be a CSV file in the below format.

<Source>, <Destination>, <Distance>

Also, The program assumes there is a windows based new line, ie there is a Carriage Return and Line feed (“\r\n”). An example view of the file (in notepad ++ with whitespaces view) is shown as follows,

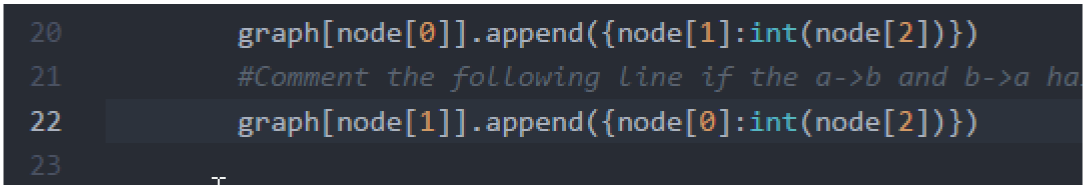


```
Oradea,Zerind,71CRLF
Oradea,Sibiu,151CRLF
```

Example:

Oradea, Zerind, 71

Note: The program considers each edge to be a 2-way edge hence the return path need not be present in the input file. In case, if there are two entries representing front and back edge, line 22 must be commented, else the program will return twice the actual distance.



```
20          graph[node[0]].append({node[1]:int(node[2])})
21          #Comment the following line if the a->b and b->a ha
22          graph[node[1]].append({node[0]:int(node[2])})
23
```

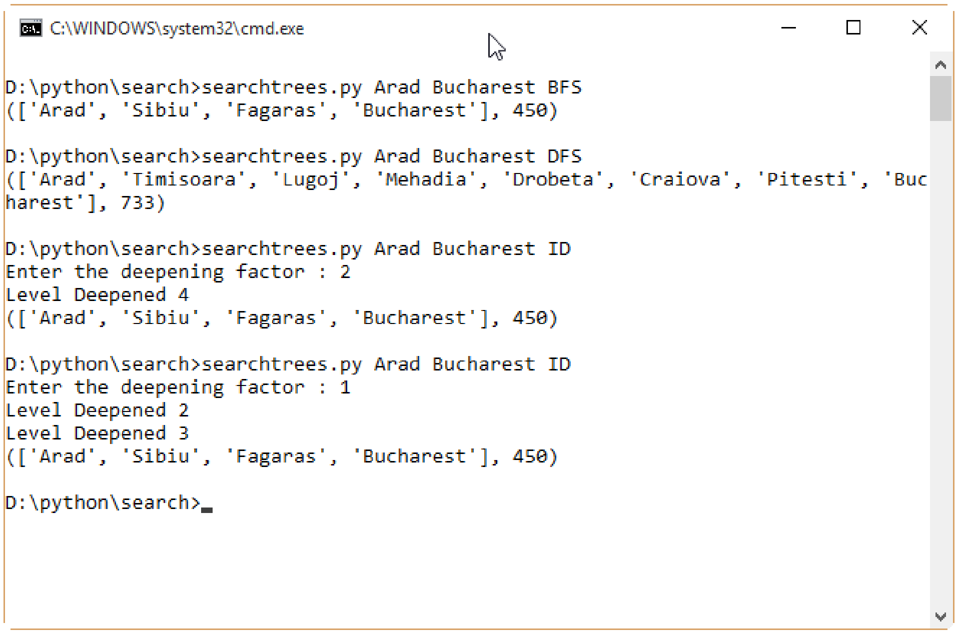
Program Execution:

The Program takes 3 parameters from the command line. The parameters should be specified in the in the below specified manner and order.

<source> <destination> <BFS/DFS/ID>

Following are the three test cases that were requested:

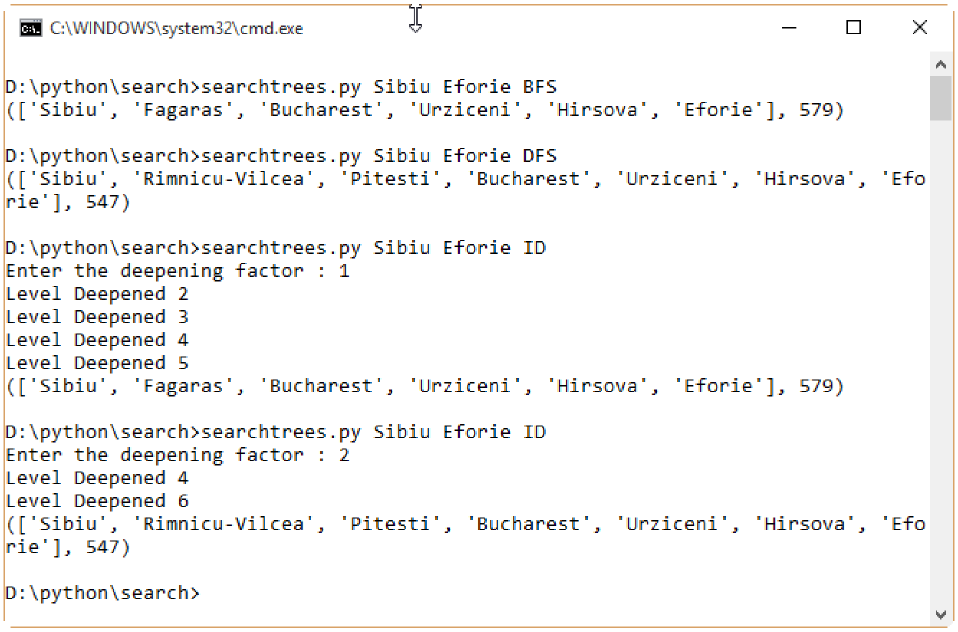
1. Arad to Bucharest:



```
D:\python\search>searchtrees.py Arad Bucharest BFS
(['Arad', 'Sibiu', 'Fagaras', 'Bucharest'], 450)

D:\python\search>searchtrees.py Arad Bucharest DFS
(['Arad', 'Timisoara', 'Lugoj', 'Mehadia', 'Drobeta', 'Craiova', 'Pitesti', 'Buc
harest'], 733)

D:\python\search>searchtrees.py Arad Bucharest ID
Enter the deepening factor : 2
Level Deepened 4
(['Arad', 'Sibiu', 'Fagaras', 'Bucharest'], 450)

D:\python\search>searchtrees.py Arad Bucharest ID
Enter the deepening factor : 1
Level Deepened 2
Level Deepened 3
(['Arad', 'Sibiu', 'Fagaras', 'Bucharest'], 450)

D:\python\search>_
```

Notice that Iterative deepening requests for the deepening factor, this is the level up to which iterative deepening must pursue from a single branch before it decides give up and search the next branch.

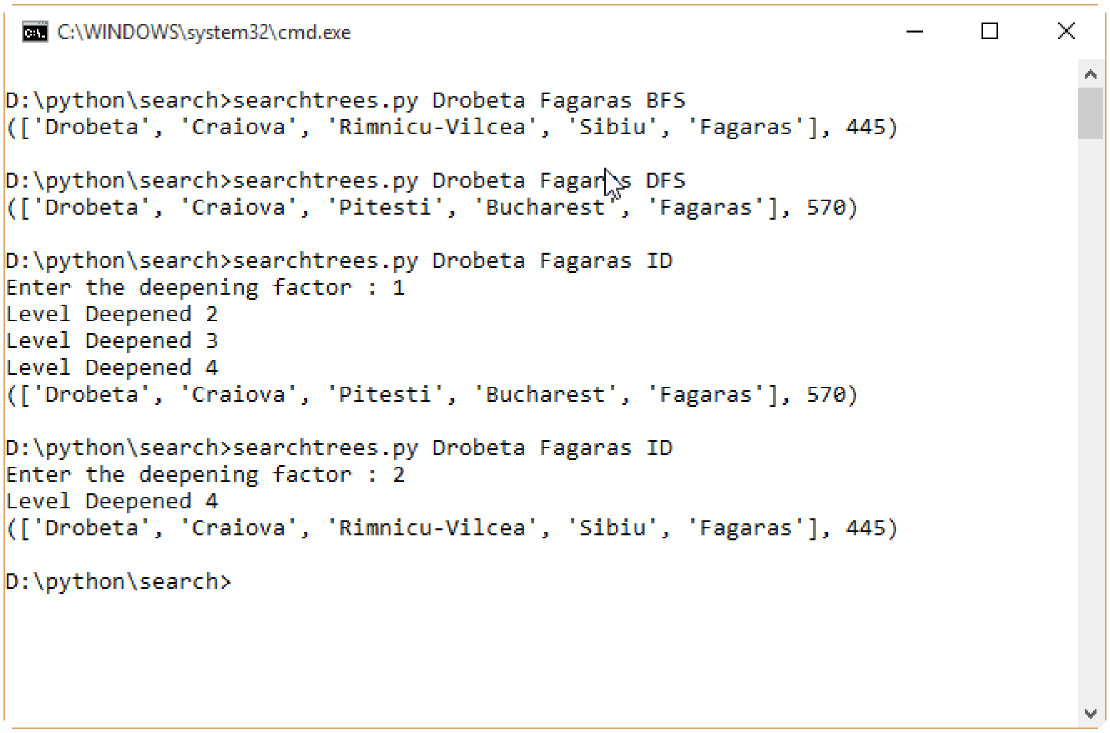
2. Sibiu and Eforie:



```
D:\python\search>searchtrees.py Sibiu Eforie BFS
(['Sibiu', 'Fagaras', 'Bucharest', 'Urziceni', 'Hirsova', 'Eforie'], 579)

D:\python\search>searchtrees.py Sibiu Eforie DFS
(['Sibiu', 'Rimnicu-Vilcea', 'Pitesti', 'Bucharest', 'Urziceni', 'Hirsova', 'Efo
rie'], 547)

D:\python\search>searchtrees.py Sibiu Eforie ID
Enter the deepening factor : 1
Level Deepened 2
Level Deepened 3
Level Deepened 4
Level Deepened 5
(['Sibiu', 'Fagaras', 'Bucharest', 'Urziceni', 'Hirsova', 'Eforie'], 579)

D:\python\search>searchtrees.py Sibiu Eforie ID
Enter the deepening factor : 2
Level Deepened 4
Level Deepened 6
(['Sibiu', 'Rimnicu-Vilcea', 'Pitesti', 'Bucharest', 'Urziceni', 'Hirsova', 'Efo
rie'], 547)

D:\python\search>
```

Notice that ID gives two results based on the depth level. One corresponds to result that is reached by BFS, other by DFS.

3. Drobeta and Fagaras:



```
D:\python\search>searchtrees.py Drobeta Fagaras BFS
(['Drobeta', 'Craiova', 'Rimnicu-Vilcea', 'Sibiu', 'Fagaras'], 445)

D:\python\search>searchtrees.py Drobeta Fagaras DFS
(['Drobeta', 'Craiova', 'Pitesti', 'Bucharest', 'Fagaras'], 570)

D:\python\search>searchtrees.py Drobeta Fagaras ID
Enter the deepening factor : 1
Level Deepened 2
Level Deepened 3
Level Deepened 4
(['Drobeta', 'Craiova', 'Pitesti', 'Bucharest', 'Fagaras'], 570)

D:\python\search>searchtrees.py Drobeta Fagaras ID
Enter the deepening factor : 2
Level Deepened 4
(['Drobeta', 'Craiova', 'Rimnicu-Vilcea', 'Sibiu', 'Fagaras'], 445)

D:\python\search>
```

As with the example before, with this use case ID gives two different results with different depth levels.