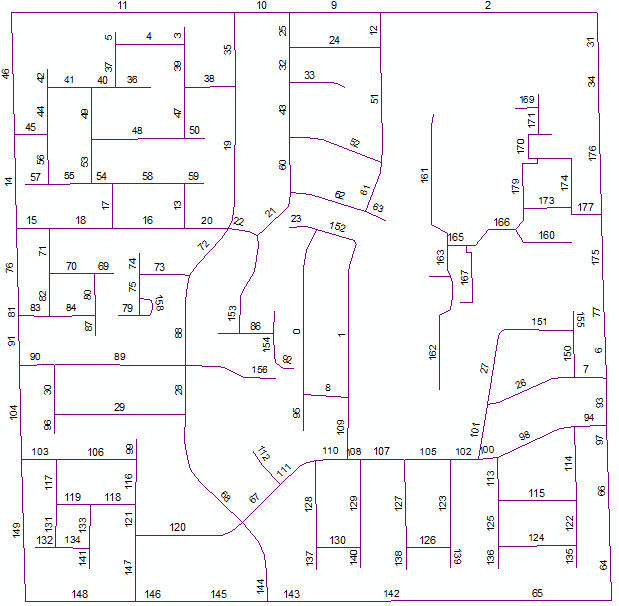
# Assignment 3: Dictionary

due: Sep 16, 11:45am

**1) Task:**

The task is to load road network data from a text file into a dictionary to subsequently compute the degree of all its nodes (intersections and cul-de-sacs). For example, the degree of node at the end of a cul-de-sac is 1, whereas the degree of a node where three roads meet is 3. The handed out .txt file contains three attributes of network arcs, which are arc feature id, the id of the FROM-node, and the id of the TO-node. The figure shows such a street network with its arc feature ids. Node ids are not shown.

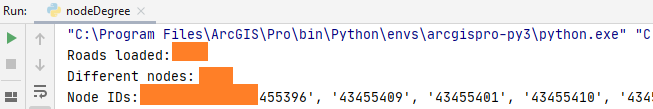


Develop a Python script which provides the following functionality:

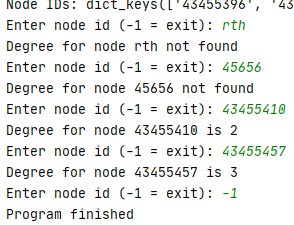
1. The program reads the streets.txt file and prints:  
   a) the number of roads loaded  
   b) the number of distinct nodes in the road network  
   c) a list of all distinct node ids in the road network
2. Next, the user is asked to enter a node id in the text box, upon which the program returns the degree of that node. If the node id is not present in the road network, a corresponding message is returned.
3. The interaction with the user continues until the user enters -1 as a node id, followed by which a “Program finished” message is returned. This should be implemented as a while loop.

**2) Example**:

The information below is displayed after loading the data (some information hidden):



The program response for a series of user inputs should look as follows:



Note the messages returned for invalid inputs, and the “Program finished” message.

**3) Implementation details:**

1) Create a new Python project folder. Copy the streets.txt file into that folder. The Python script you develop should be located in the same folder as well (no subfolder or parent folder).

Do **not use absolute paths** to point to the street.txt file (such as C:/data/streets.txt). Instead, use paths relative to your python script, based on functions offered in the *pathlib* module

2) Your scripts should create an empty dictionary called dict.

3) Then use a method of your choice to read in the road data line by line from the streets.txt file, using *pathlib* module functions.

4) Extract both node ids of each arc (i.e. FROM-node id and TO-node id) when looping through the lines. Note the that the first line in the .txt file contains only header information which should be ignored.

5) Use node id (string type) as a key for the dictionary, and the node degree as the value. For each extracted node id (whether it’s a FROM-node or TO-node), check whether that key already exists in the dictionary.

* If this is not the case, add a new key:value pair with key = node id, and value (node degree) = 1 to the dictionary.
* If the key already exists, increase the current value of that key:value pair by one.

Once the program ran through all lines, this procedure provides the node degree for each node.

Here is the suggested structure of your program:

**from** pathlib **import** \*

... # specify path information to streets file  
...

dict = {}

... # read-in text file content

... # split content into individual lines

**for** line **in** lines:

...# obtain elements from each line

...

...**if not** node **in** dict.keys():

... # handle distinct node count

...

print(**"Roads loaded: "** + str(roadnr))

...

id = 0

while id != -1:

id = input(...)

...

**if** id **in** dict.keys():

...

else:

...

print(**"Program finished"**)

**4) Program execution:**

a) Run your program in PyCharm;

b) Use 555, 876612407, 43444392, 43444482, and -1 as input node ids;

c) Insert a screen capture of the console window (which should look similar to the example above).

< HERE >

Hand in:

A zip file containing:

1) This MS Word document with the inserted screen capture.

2) A Python script file with your code.

Good luck!