Minimum Spanning Tree of a Graph

Homework #13
By
John Robertson
CS 303 Algorithms and Data Structures
November 26, 2019

1. **Problem Specification**

This problem requires us to extend our graph model to create a minimum spanning tree out of it.

2. **Program Design**

This program has two files: a file with a node and graph class and a driver file.

The following steps were required to develop this program:

- Make minor adjustments to the Node and Ugraph classes
- Implement MST-PRIM, EXTRACT-MIN, and a method to display the minimum-spanning tree
- Debug the above until they worked, writing a mini-driver within the file.
- Make minor adjustments to the driver of the last two assignments to accommodate weights

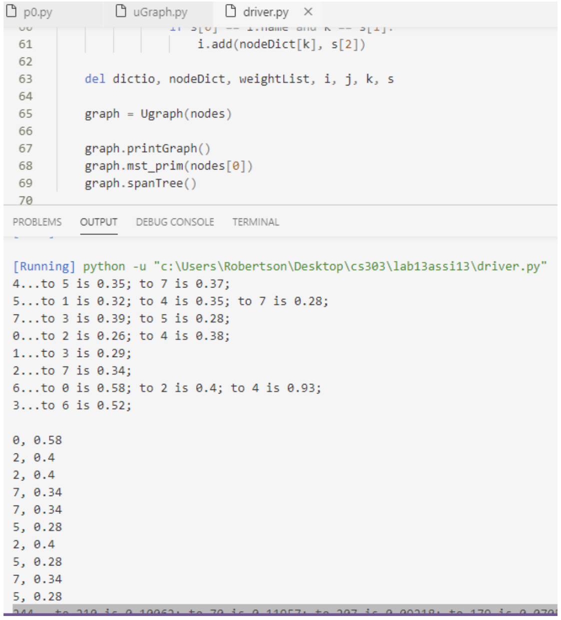
3. **Testing Plan**

The mini-driver manually populates a graph with nodes and their connections. It simply tests running MST-PRIM by running and using printGraph() and then spanTree().

4. **Test Cases**

The driver tests the files in sequence to ensure that the unpack itself works correctly. I had initially forgotten to call MST-PRIM, leading to spanTree() mysteriously not working until I realized it. Its output is similar to the output for the mini-driver from Ugraph.py, which.

5. **Analysis and Conclusions**



```
                if s[0] == i.name and k == s[1]:
                    i.add(nodeDict[k], s[2])

    del dictio, nodeDict, weightList, i, j, k, s

    graph = Ugraph(nodes)

    graph.printGraph()
    graph.mst_prim(nodes[0])
    graph.spanTree()
```

```
[Running] python -u "c:\Users\Robertson\Desktop\cs303\lab13assi13\driver.py"
4...to 5 is 0.35; to 7 is 0.37;
5...to 1 is 0.32; to 4 is 0.35; to 7 is 0.28;
7...to 3 is 0.39; to 5 is 0.28;
0...to 2 is 0.26; to 4 is 0.38;
1...to 3 is 0.29;
2...to 7 is 0.34;
6...to 0 is 0.58; to 2 is 0.4; to 4 is 0.93;
3...to 6 is 0.52;

0, 0.58
2, 0.4
2, 0.4
7, 0.34
7, 0.34
5, 0.28
2, 0.4
5, 0.28
7, 0.34
5, 0.28
```

I have managed to increase the inefficiency of my file unpacking algorithm by making it O(n^3). This is an example of trade-offs: if I had the time to invest in finding a better algorithm, it could probably be done in O(n). As for outputting a minimal spanning tree… I have failed to do this. Although my implementation of MST-PRIM may be correct, I have no way of verifying this other than that it causes no errors. The problem is likely in my implementation of EXTRACT-MIN, which I had to create from scratch.

6. **References**

Node and weight values provided by Logan Creel

https://stackoverflow.com/questions/59003032/how-do-you-iterate-through-a-queue-in-python-without-emptying-it?noredirect=1#comment104256465_59003032

https://realpython.com/iterate-through-dictionary-python/#iterating-through-items