Andrew Breyen
Programming Assignment 1 Report

Approach

(this information is also commented well in the toposortv2.py file)

Global:

I used an array to hold Boolean values if nodes were visited or not

Method topoSort:

The method topoSort takes a visitedVertex, the graph, and a specific vertex. It creates a python stack (basically just an array) to hold the order

visitedVertex is set to the result of the DFS.

For every vertex in the graph, if vertex is not visited, so a DFS with the parameters visitedVerted, the graph, the specified vertex, and the stack.

At the end of the for loop, the stack is returned.

The topoSort method is what is called to actually do the topological sort from the main method.

Method DFS:

The method DFS takes the visited vertexes array, the graph, the vertex, and the stack.

First, the current vertex is appended to the visited vertex array.

Then, if the specified vertex is in the graph:
For every element in the graph at the specified vertex if it has not been visited, do another (recursive) call of DFS with the visited array, the graph, and the element (in this case it is w)

After the if statements are done, the specified vertex (v) is appended to the stack.

Then, visited is returned.

"Main" Method:

First, the file from the command line arguments is processed and the lines are read into lines.

Next, a 2D array is created to hold the vertexes and the target vertexes.

After that, for line in lines, the vertexes and vertex targets are appended to the vertex and target vertex 2D arrays respectively.

Next, an empty graph is created. This is created as a dictionary.

After the graph/dictionary is created, for all the lines in the text file, v is set to the vertex array at position i.

Then, if the vertex is in the graph, graph[v] is set to the target vertex.
If it is not in the graph, it is also set to targetVertex[i].

One thing I found interesting here is the "|=" operator which takes into consideration all nodes connected to first node. Without it, it just considers the last connected node.

Next, a time is logged. This is the start time.

doTheSort is printed, and the result of the topoSort method is set to topoOrder.

After the sort is complete, elements are popped off the stack and printed to represent the result.

The timer is then stopped, total execution time calculated, and the total execution time is printed.

Things to Know

When testing my makefile/tshort.sh files, I got different results, even though I used the correct topoSort.py file in my tsort.sh file.

Results from VS Code when executing the shell script directly:

```
Andrews-MacBook-Pro: abreyen$ python toposortv2.py 5.ij

doTheSort!
2
1
3
4
5
Execution Time in Seconds: 5.19752502441e-05
```

Results when running `tsort 5.ij` directly from the terminal in my 900231310 folder:

```
Andrews-MacBook-Pro:900231310 abreyen$ tsort 5.ij

2
1
3
4
5
```

The sort is completing successfully in the 900231310 folder, but is not printing the “doTheSort” text nor the execution time text. I am not sure why this is happening, but I have verified that I used the correct python file, and I believe that I created the makefile and tshort.sh correctly.

Real World Problem

Topological sorting can be helpful when scheduling classes that have prerequisites. Some classes depend on the completion before you can take higher level courses (150 must be taken before 160 before 310 etc)