

CptS 223 Extra Credit – Topographical Sort

In this task you'll implement Topographical Sort. This sort algorithm is laid out in chapter 9 of our book. I've included code to build the graph from a GraphViz dot file. This is based on our code from the Dijkstra's MA.

Because toposort doesn't specify order when there are multiple nodes with an indegree of zero, I can't pre-test your output, but I'll know it when I see it done properly.

Here are some of the key requirements:

- 1) Your toposort algorithm must return a list of vertex pointers
- 2) Your toposort algorithm must actually remove the vertices from the graph as you execute
- 3) The code will print the graph after you run toposort and it should have ZERO nodes left

Yes, you'll need to modify the Vertex objects, maybe change how vertices are stored in the graph and eventually just chuck everything but my main and the dot file parser. That's entirely up to you.

The points for this assignment will count towards your PA/Homework pool of points.

*** NOTE: This code DOES NOT run on the new sig servers ***

It's based on the same regex parser from the Dijkstra's MA so it won't run and I don't have time to fix it right now. You MUST test your code on the OLD SSH servers: `ssh1.eecs.wsu.edu`

The code is in a Git branch is called: EC1

Testing

The final tests basically go:

- 1) load dot file into graph
- 2) print graph out
- 3) run toposort
- 4) print toposort results
- 5) print graph out (with no nodes since toposort destroyed it)

Deliverables

You must commit your code to your Git repo and put the file with your git commit hash on Blackboard, as per usual.

Grading Criteria

Your assignment will be judged by the following criteria:

- [16] Final tests output a proper toposort
- [2] Data structure usage. Your program actually deletes the graph as it runs toposort
- [2] Your code is well documented and generally easy to read.