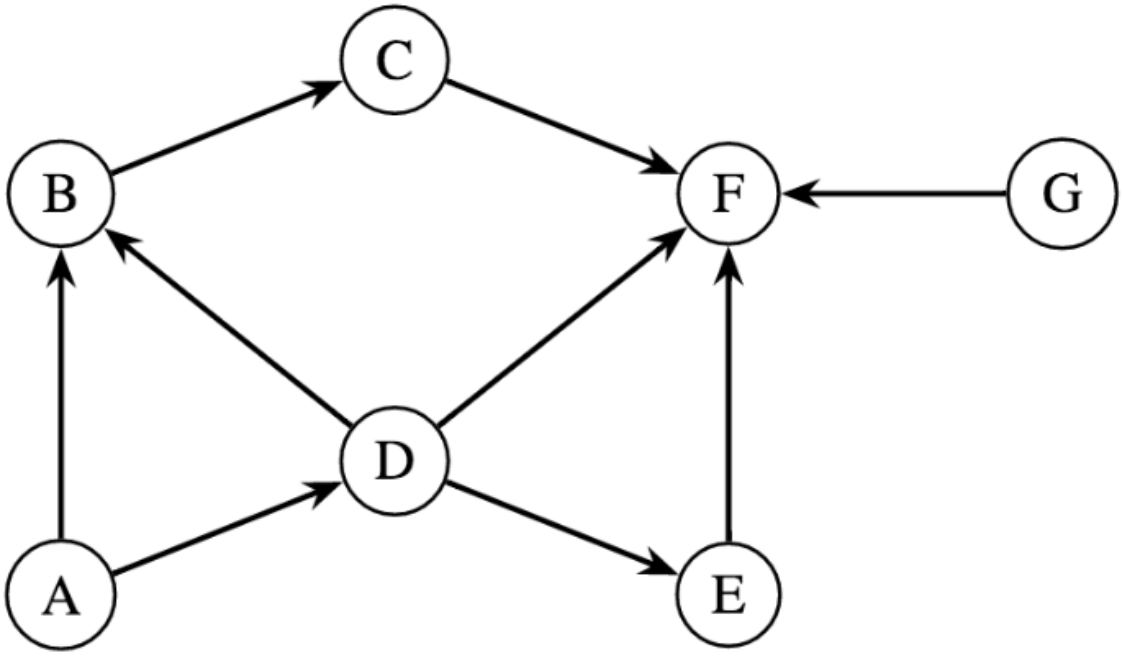
TA: Sherdil Niyaz Wednesday, April 6, 2016

Discussion 11 Correction

Topological Sorting



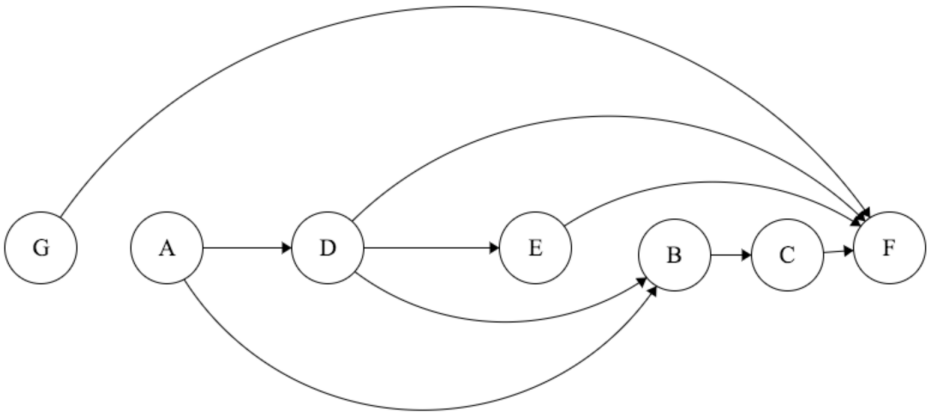
In discussion 11, we saw this graph. Problem 3 asked for a valid topological sort, so we took the post order from DFS:

FCBEDA

and reversed it:

ADEBCF

This gave us the following topological sort (see next page:)



G didn't appear in our post order since it was not *reachable* from A, but we just have to make sure it appears before F since that's the only node it has an edge out to.

Soooo….

Why does reversing the post order work? (This is the point that I butchered so terribly in section!) Here's an intuitive argument. You'll see a more rigorous version in CS 170.

Remember that, in a topological sort, the edges always have to go from the left to the right. Remember also that we also **reversed** the post-order to get the topological sort given above.

For the first node that appears in our post-order, we note that this is the **first** node that the DFS function returned on. What does this mean? Well, assuming that the graph is not cyclic, there can be no other edges directed out of this node! If there were, we would have called DFS on the nodes that those edges led to, which would make this node actually *not* the first in the post order….which we know isn't true (this is a mini proof by contradiction if you're in CS 70. If you aren't, don't worry about the meaning of that term.)

So if we put this node **last** in our topological sorting, then we know that edges will only go **into** it, keeping a valid topological order. Note that we can see this in F above!

Let's take this idea and generalize the intuition to any node in the graph! If there is an edge to some node "deeper" in the graph (farther away from A), it will have DFS return on it earlier than some "shallower" node that leads to it, since the DFS call on the "shallower" node has to wait for the "deeper" node's call to finish. You can see this from the relationship between D and E in this problem, for example.

This is why reversing the post order works to give us a valid topological sort! The "deeper" nodes have their DFS calls return earlier, and thus end up earlier in the post order. We want those "deeper" nodes to be **later** in our topological sort, though (to avoid edges going the wrong way) so we flip the post order to get the actual Topological Sort.

A final note: I made a point that this algorithm (flipping the post order) won't work if you just applied it blindly. What this means is that this algorithm obviously won't work if the graph is cyclic, since you can't Topologically Sort a cyclic graph! But the algorithm doesn't know to throw its hands up and say "Hey! I can't sort this!"….You have to check for cycles on your own!

Feel free to shoot me an email using sniyaz *at* berkeley.edu if you have more questions. Apologies again for the mix up. I am also more than happy to meet one-on-one to explain this if you do not understand.