CSCI 230 -- PA 10

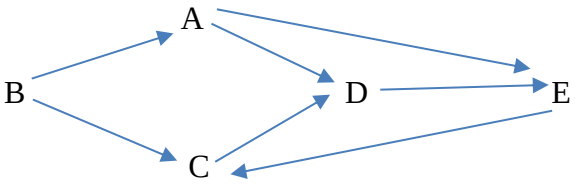
Digraphs and Some Graph Algorithms

Feel free to discuss and help each other out but does not imply that you can give away your code or your answers! Make sure to read all instructions before attempting this lab.

You can work with a lab partner and each one must submit the same PDF file (include both names in the submission file). Each person must include a brief statement about your contribution to this assignment.

You must use an appropriate provided template from Canvas and output "Author: Your Name(s)" for all your programs. If you are modifying an existing program, use "Modified by: Your Name(s)".

Use the following digraph to test exercise 1 and exercise 2.



Exercise 1: Implement Transitive Closure for a digraph using AdjacencyListGraph class from previous PA. Test it out on a simple example above (can assume each edge has a weight of 1). Print both original digraph and updated digraph.

Exercise 2: Implement a shortest path algorithm for a graph (preferably Dijkstra's algorithm) using AdjacencyListGraph class from previous PA. Test it out using digraph above. Find the shortest path from B to E using the weights below. Print out the path how to get from source vertex to a destination vertex. *Hint: need to store the vertex that you use to update the distance and you can backtrack from destination back to source vertex.*

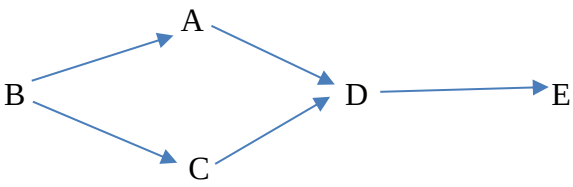
- <B, A>, 3
- <B, C>, 4
- <A, D>, 5
- <A, E>, 10
- <C, D>, 2

- <D, E>, 3
- <E, C>, 6

Question 1: What is a DAG? Give an example of a real-life DAG.

Question 2: Explain the concept of "Transitive Closure" on a digraph. Provide a reason for performing a transitive closure on a digraph.

Extra Credit: Implement **Topological Ordering** on a DAG using AdjacencyListGraph class from previous PA. Test it out on the following DAG:



Fill out and turn in the PA submission file for this assignment (save as PDF format).