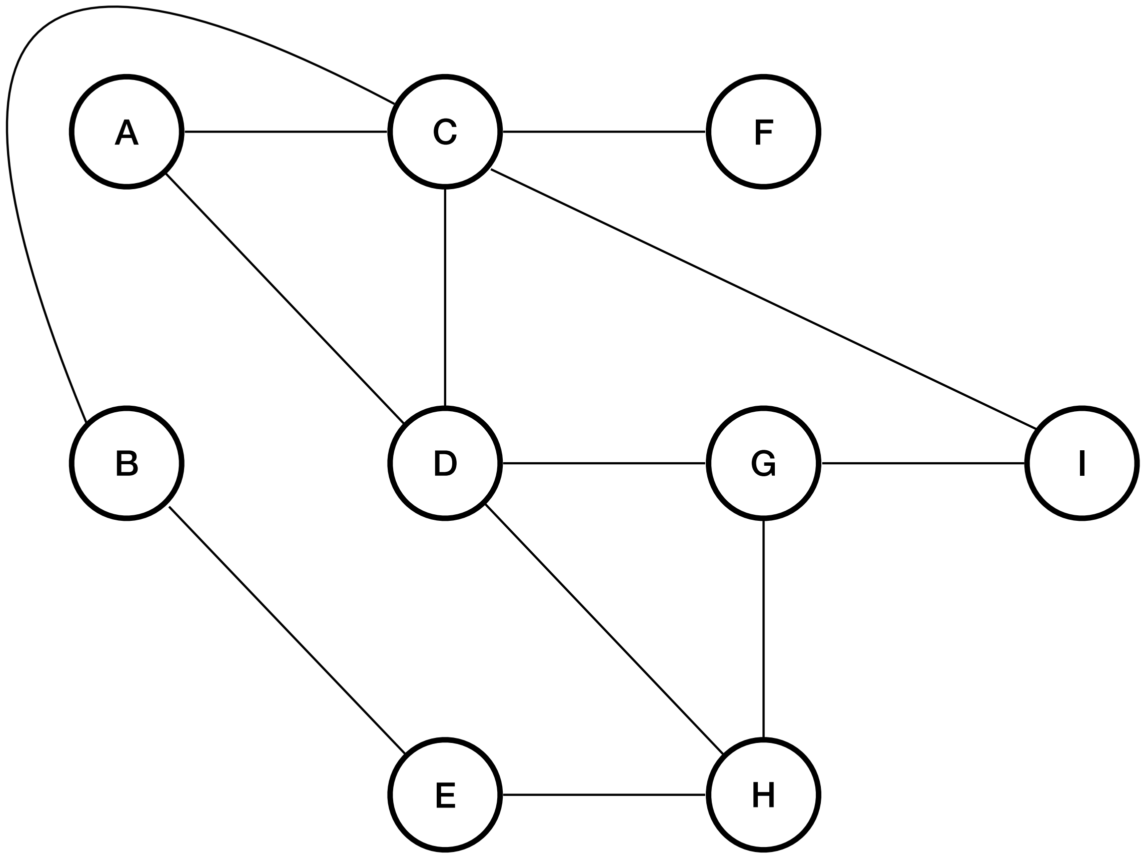
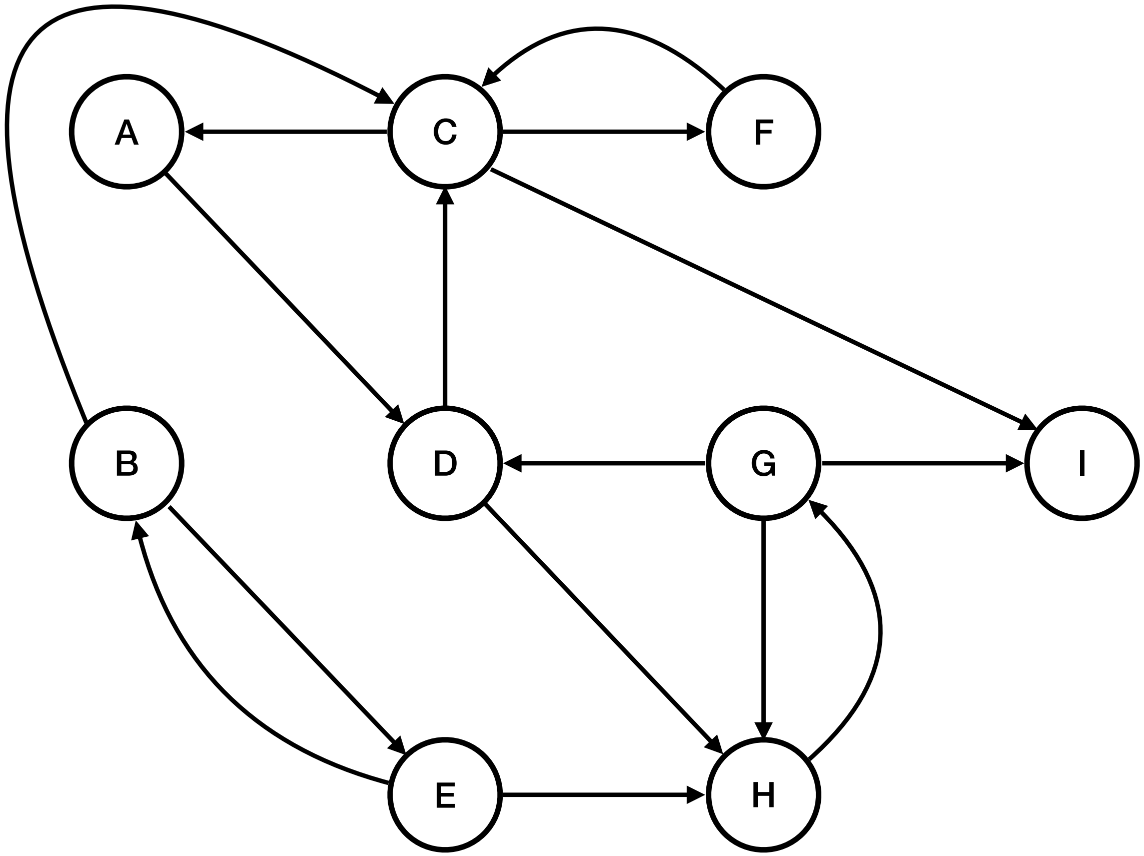
1. Undirected Graph



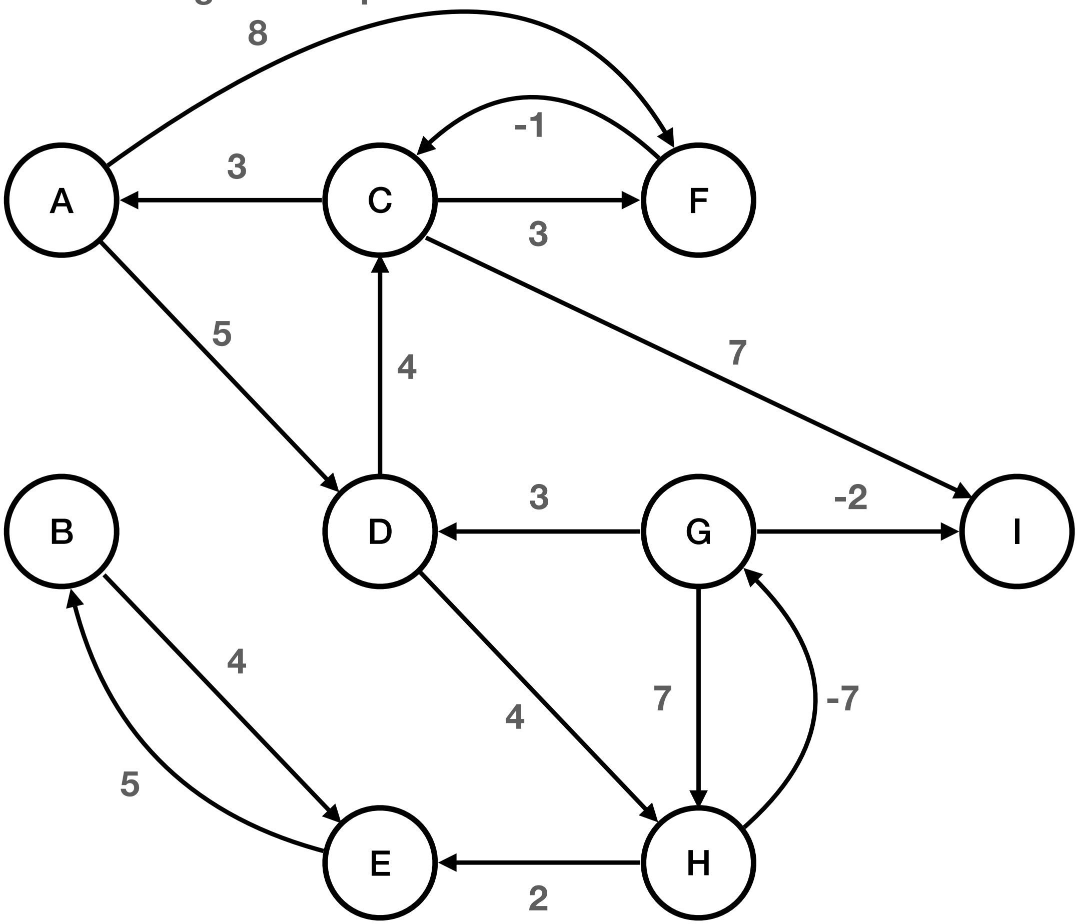
1. Run DFS on the graph starting at node A and show the exploration tree, including pre/post values. Process vertices in alphabetical order.
2. Run BFS on the graph starting at node A. Show the exploration tree.

2. Directed Graph



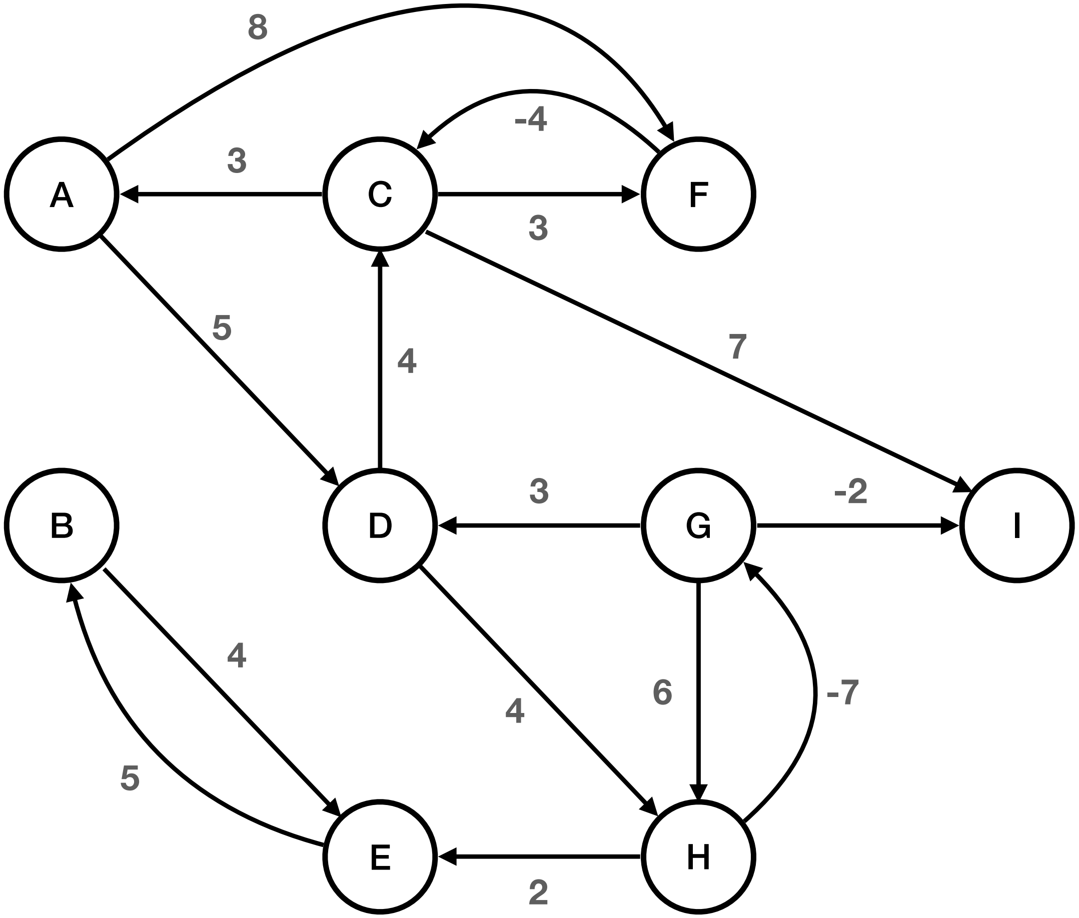
1. Starting with node I, run DFS on the reverse graph, storing pre/post
2. Run connected components on the original graph, initialize the DFS search using the vertices in decreasing order of their post numbers. Once a vertex is included in a component, it cannot be added to another. Show the connected components and the meta-graph giving the connectivity of each group. Identify which groups are sources and which sinks.

3. Directed Weighted Graph



1. Create the DFS tree, starting at node A. Show all forward, back, and cross edges and pre/post numbers. How do you know that back edges exist in the graph? The existence of a back edge implies what?
2. Execute Dijkstra's Algorithm on the graph using the source node A, showing the resulting spanning tree and distances. Does it work on this specific graph with negative edges? Why, or why not?

4. Directed Weighted Graph



1. Execute the Bellman-Ford Algorithm on this graph, showing 9 iterations in a table like the one below. Does the graph have a shortest path solution?
2. What is different about this graph versus the prior graph?

N	A	B	C	D	E	F	G	H	I
0									
1									
2									
3									
4									
5									
6									
7									
8									
9									