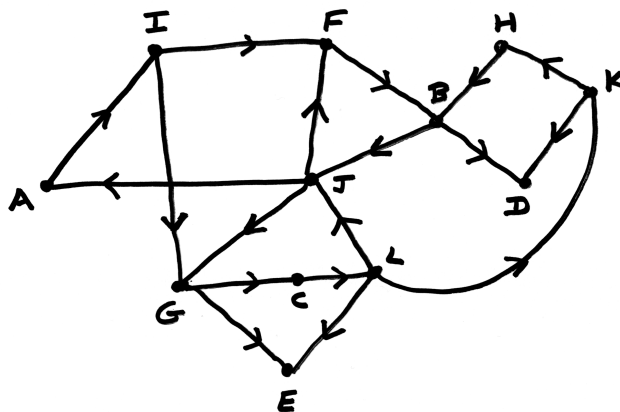
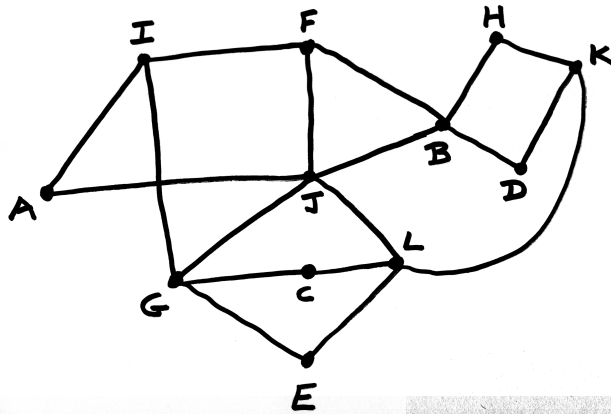


### Shortest Paths (Dijkstra's Algorithm)

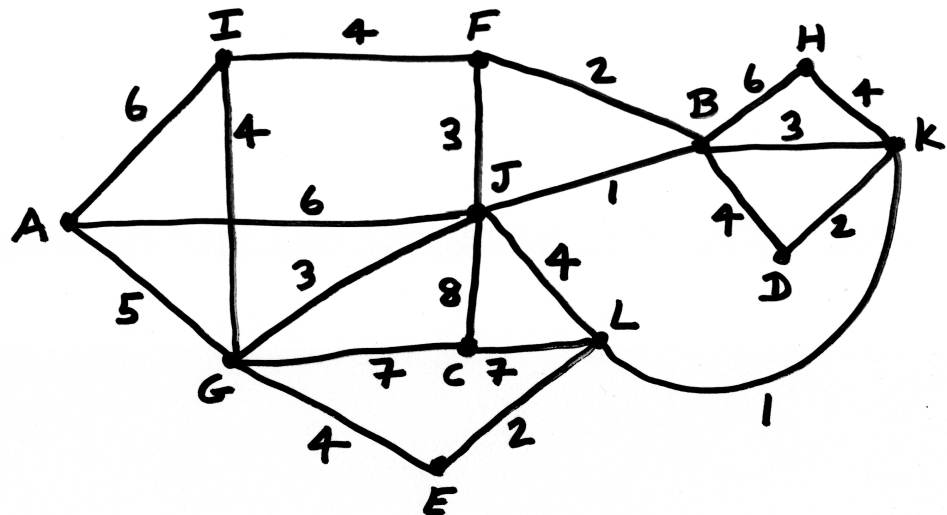
- For each of the graphs below (one undirected, the second directed) find the shortest distances from vertex  $A$  to all other vertices. (Note that the edges  $\{I, G\}$  and  $\{A, J\}$  cross each other, but there is not a vertex at the point of intersection). For each graph, draw the subgraph that consist of edges that are used in the shortest paths.

You should find *both* the shortest distances *and* the predecessor array which will allow us to reconstruct a path joining  $A$  to any vertex.

For each of these two graphs, the weight of every edge is 1 (and hence, that's why I haven't included the weights in the diagram).



2. Consider the weighted (undirected) graph below. (Edges  $\{G, I\}$  and  $\{A, J\}$  cross but there is not a vertex at their intersection.)



Find shortest paths from vertex  $A$  to all other vertices in the graph. You should find *both* the shortest distances *and* the predecessor array which will allow us to reconstruct a path joining  $A$  to any vertex.

As above, *draw* the subgraph that contains only those edges used in the shortest paths.