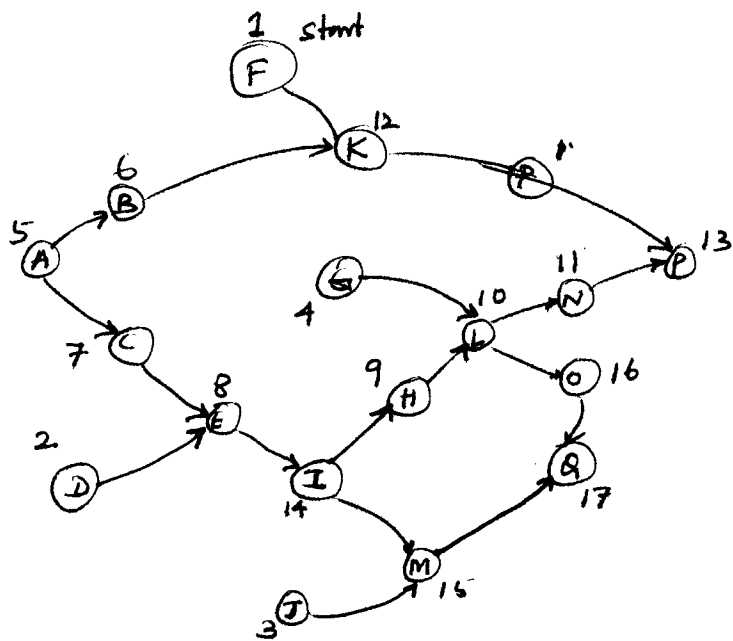


HW#4 SOLUTIONS

Q1:

- a) • If no vertex has in-degree zero, then the graph is cyclic because it does not have a source.
- If the graph is acyclic, then we will be able to start the first step. After assigning vertices with positive labels we see new sources emerge. (unlabeled vertices) Then we mark them and remove over and over till we label all the vertices.
 - If a edge (u, v) exists, then $d(u) < d(v)$ because unless u is positive, we cannot have put a label on v , thus $d(u) < d(v)$. Also for any path from u to v , u has to have a label less than v . Thus we get a topological sort by labelling the vertices this way.

b)



The sorted array looks:

F D J G A B C E H L N K P I M O Q.

c)

Step 2: $O(V)$

Step 3, 4: $O(V^2)$

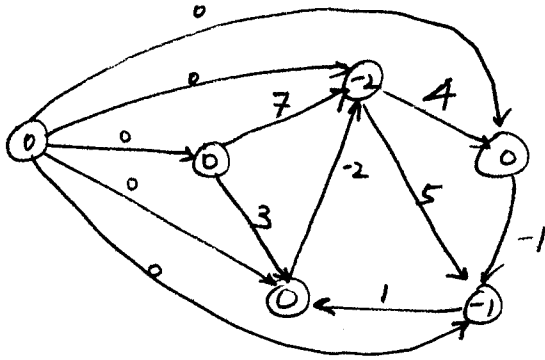
Effectively $O(V^2)$ or $O(V \cdot E)$

Worstcase $O(V^3)$

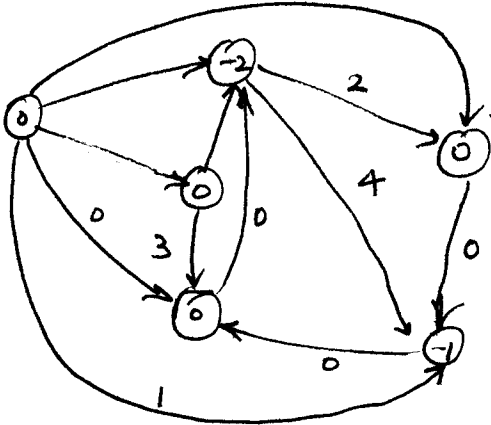
Q2: Requirement: A Successfully compiling program

+
3-4 Examples showing topologically sorted arrays.

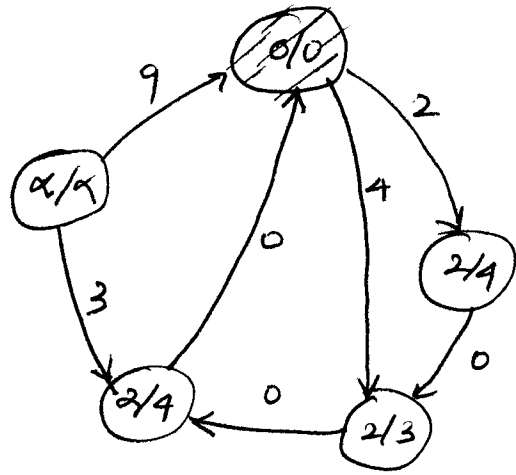
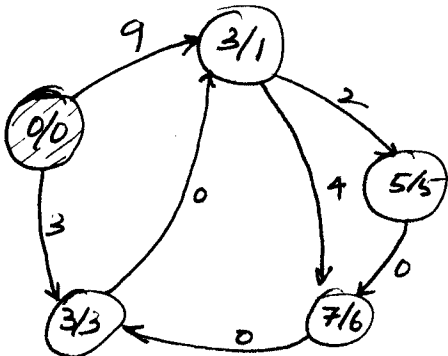
Q3: Johnson's Algo.

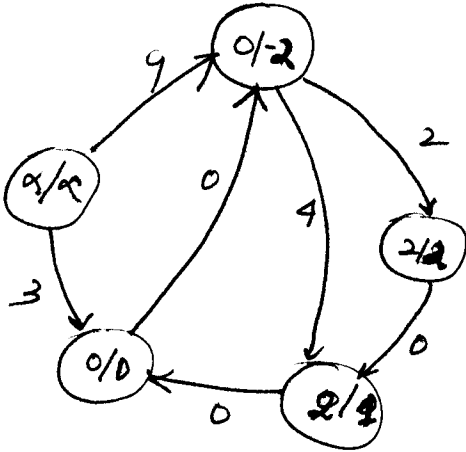
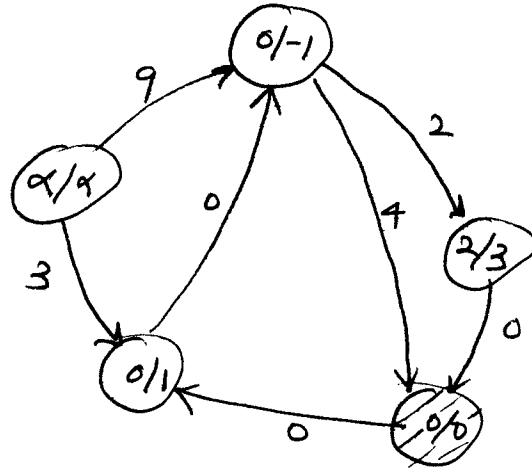
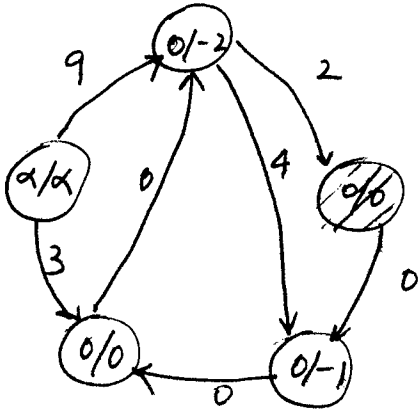


Bellman-ford



Dijkstra's at each node





b) The algorithm just flashes/Reports an error message saying that the "graph contains a negative cycle".