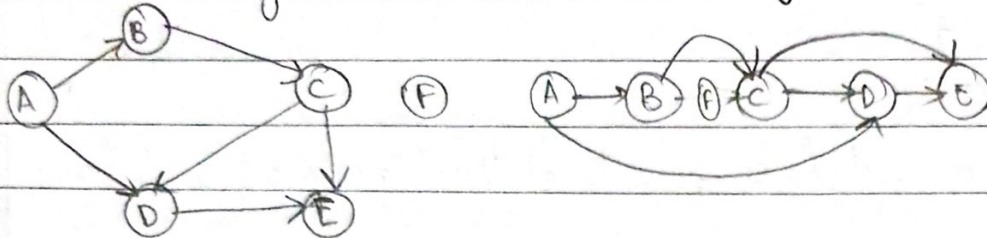


14. TOPOLOGICAL SORT

Topological sorting for a directed acyclic graph (DAG) is a linear ordering of vertices such that for every directed edge uv , vertex u comes before v in the ordering. Topological sorting for a graph is not possible if it isn't a DAG.

Any linear ordering in which all the arrows go to the right is a valid solution.

eg:



Applications:

- (i) Scheduling jobs from the given dependencies among jobs.
- (ii) Instruction scheduling.
- (iii) Determining the order of compilation tasks to perform in makefiles.
- (iv) Data serialization.

Algorithm:

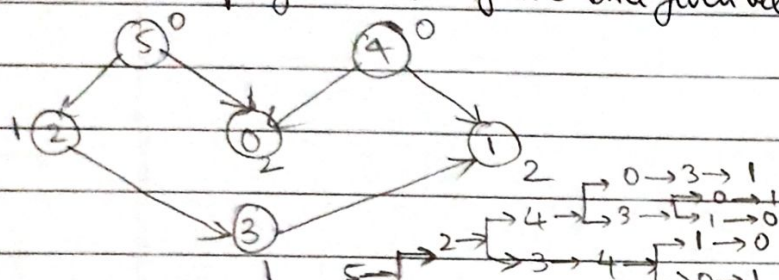
1. Identify vertices that have no incoming edge.
 - The in-degree of these vertices is 0
 - If no such edges, the graph has cycles (cyclic graph)
 - Select one such vertex (with in-degree 0).

2. Delete this vertex of indegree 0 & all its outgoing edges from the graph.
Place it in the output \rightarrow In degree of all other vertices \downarrow by 1.
3. Repeat steps 1 & 2 until the graph is empty.

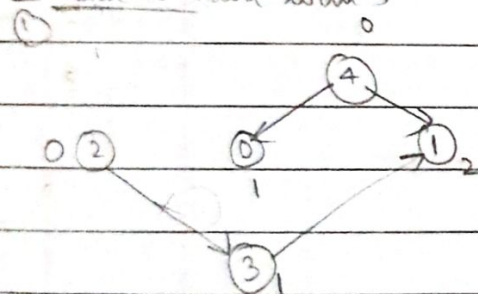
OR:

1. store the indegrees of all vertices in an array.
2. while there are vertices remaining:
 - Find a vertex with indegree 0 & output it
 - Reduce the indegree of all vertices adjacent to it by 1
 - Mark this vertex (In-degree = -1).

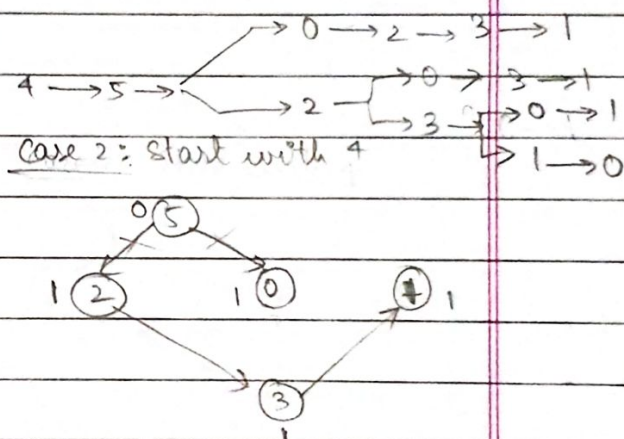
a. Find the topological sort of the DAG given below:



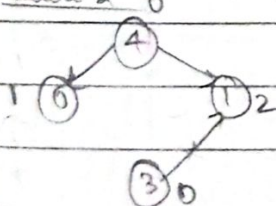
I. Case 1: Start with 5



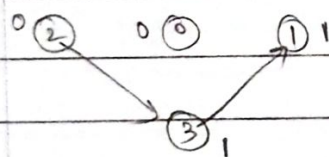
Case 2: Start with 4



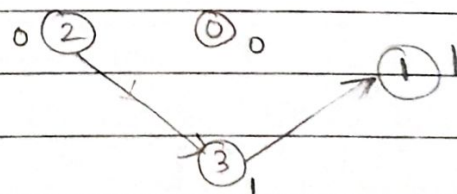
Delete 2



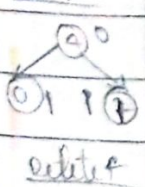
Delete 4



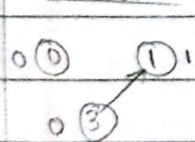
Delete 5



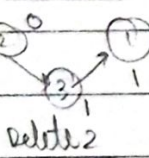
Delete 3



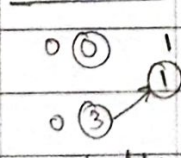
Delete 4



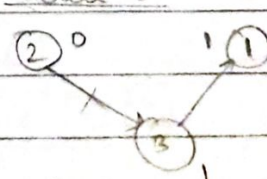
Delete 0



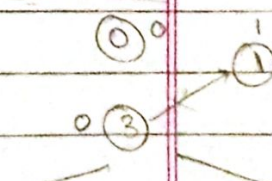
Delete 2



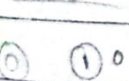
Delete 0



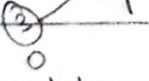
Delete 2



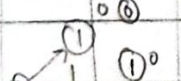
Delete 4



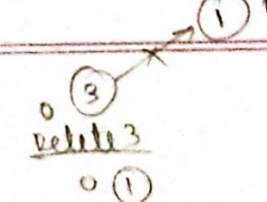
Delete 3



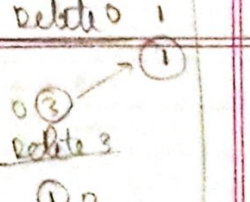
Delete 0



Delete 2



Delete 0



Delete 3

