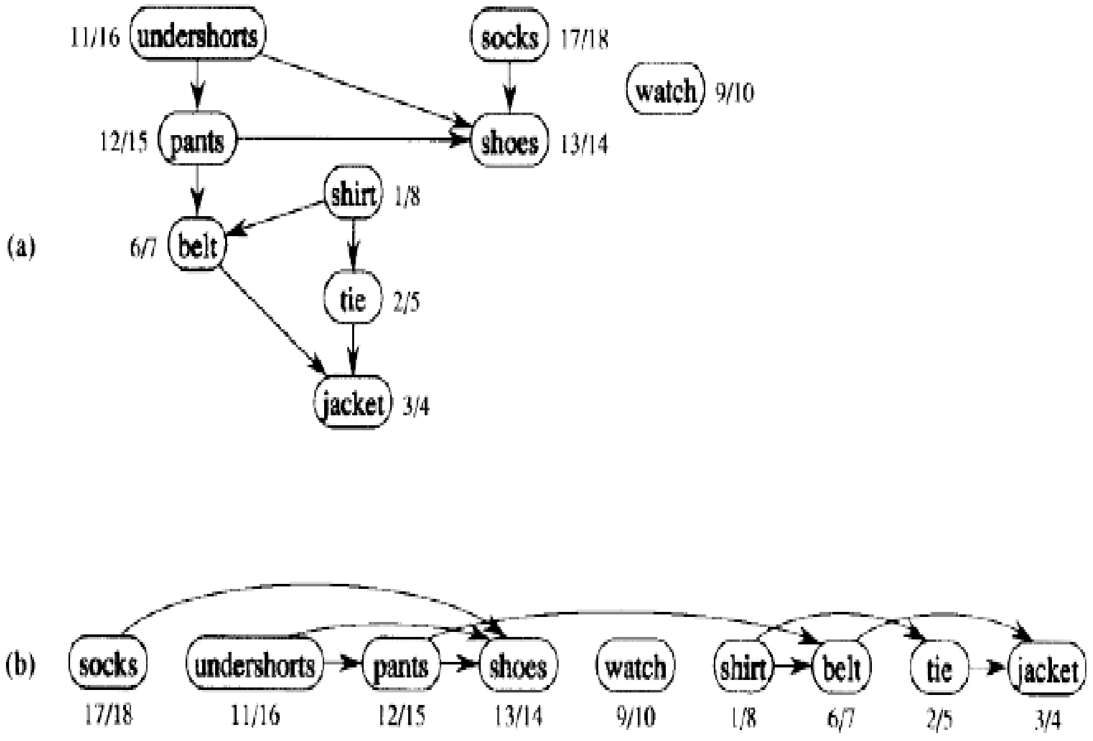
TOPOLOGICAL SORTING

- A cycle in a diagraph or directed graph G is a set of edges, $\{(v_1, v_2), (v_2, v_3), ..., (v_{r-1}, v_r)\}$ where $v_1 = v_r$. A diagraph is acyclic if it has no cycles. Such a graph is often referred to as a directed acyclic graph, or DAG
- Topological sorting for Directed Acyclic Graph (DAG) G =(V,E) is a linear ordering of all its vertices such that if G contains an edge (u,v), then vertex u comes before v in the ordering.
- Topological Sorting for a graph is not possible if the graph is not a DAG.
- We can view a Topological Sort of a graph as an ordering of its vertices along a horizontal line so that all directed edges go from left to right.

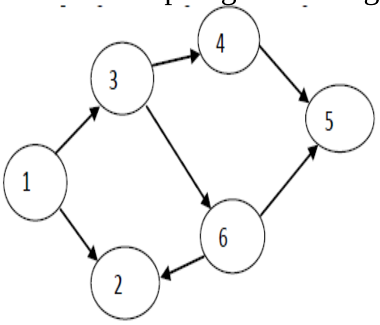
Algorithm

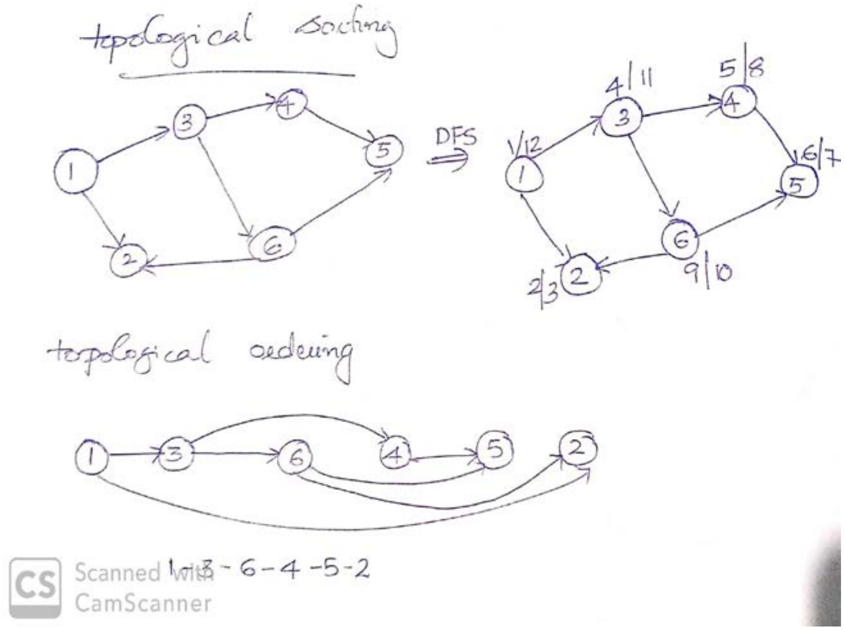
```
TOPOLOGICAL-SORT(G)

1. Call DFS(G) to compute finishing times f[v] for all v in V
2. as each vertex is finished, insert it onto the front of a linked list
3. return the linked list of vertices
```

- **Time complexity**: We can perform a topological sort in time , O(V+E) , since depth-first search takes O(V+E) time and it takes O(1) time to insert each of the V vertices on to the front of the linked list.



1.Write the topological sorting for the DAG given below [December 2019-2marks]




topological sorting

topological ordering

1-3-6-4-5-2

topological sorting 1-3-6-4-5-2