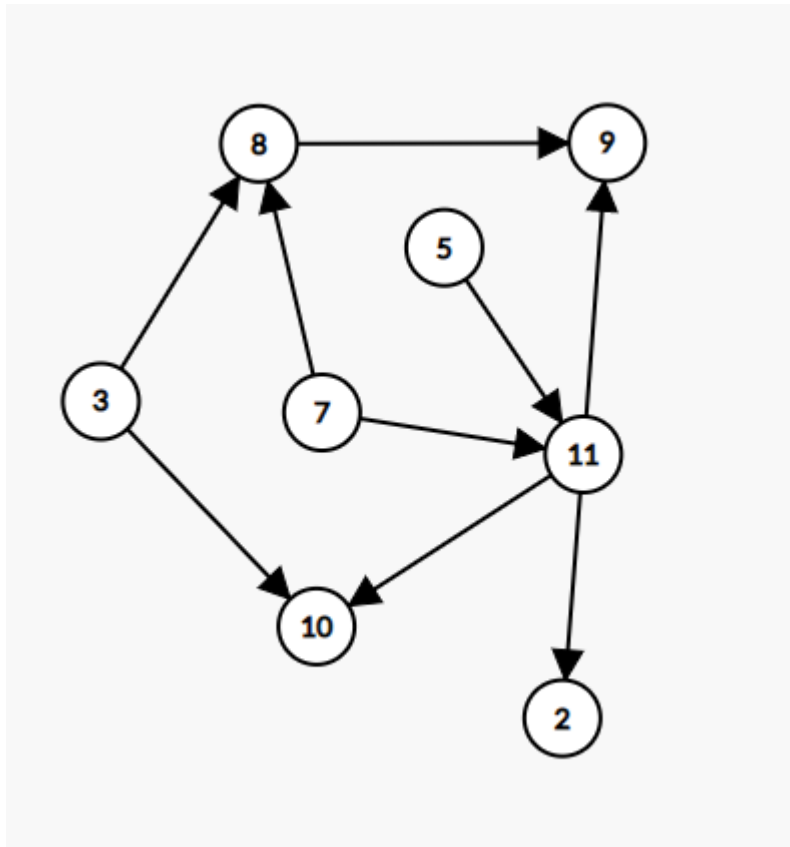# Topological Sort

Topological sort is an ordering of vertices in a directed acyclic graph in which each node comes before all the nodes to which it has outgoing edges. As an example consider the course prerequisites structure at universities. A directed edge(v, w) indicates that course v must be completed before course w. Topological ordering for this example is the sequence that does not violate the prerequisite requirement. Every directed acyclic graph must have one or more topological ordering. Topological Sort is not possible if the graph has a cycle since, for two vertices v and w in the cycle, v precedes w and w precedes v.

The topological sorts for above graph are: 7, 5, 3, 11, 8, 2, 9, 10

3, 5, 7, 8, 11, 2, 9, 10

Algorithm (Kahn's algorithm):

1. Indegree is computed for all vertices, starting with the vertex having indegree 0.
2. To keep a track of vertices with in-degree 0 we can use a queue.
3. All vertices of in-degree 0 are placed in a queue.
4. While the queue is not empty, a vertex v is removed and all vertices adjacent to v have their indegrees decremented.
5. A vertex is put on the queue as soon as its indegree falls to 0.
6. The topological ordering is the order in which the vertices dequeue from the queue.

Time Complexity: O(|E| + |V|)

Application of Topological Sort:

- Representing course prerequisites
- In detecting deadlocks
- The pipeline of computing jobs.
- Checking for symbolics link loop
- Evaluating formulae in a spreadsheet