

# L22 - Kahn's Algorithm

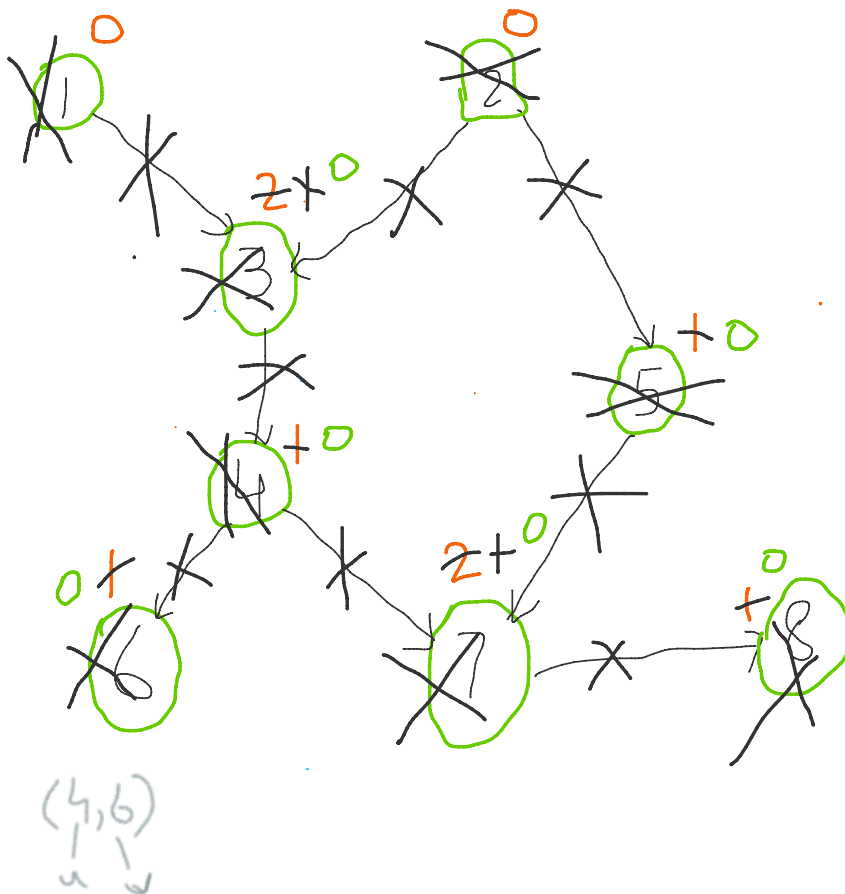
Monday, June 22, 2020 3:59 PM

Topological Sorting of a DAG is Linear Ordering of its vertices such that for an edge  $(u, v)$ , the vertex  $u$  appears before vertex  $v$ .

## Kahn's Algorithm

In-degree of a node: Number of edges coming/terminating at a node

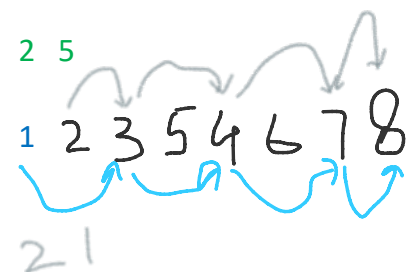
Out-degree of a node: Number of edges originating from a node



1 2 3 4 5 6 7 8

## Adjacency List Representation

1	3	
2	3	5
3	4	
4	6	7
5	7	
6		
7	8	
8		



1. Let inDegree be an array of size  $|V|$
2. For each  $u \in G.V$
3.      $U.inDegree = 0$
4. For each edge  $(u, v) \in G.E$
5.      $V.inDegree++$

$O(|V| + |E|)$

1. Let inDegree be an array of size  $|V|$
2. For each  $u \in G.V$
3.      $inDegree[u] = 0$
4. For each  $u \in G.V$
5.     for each edge  $adjMat[u][v]$
6.          $V.inDegree++$

1. Calculate the in-degrees of every vertex  $O(|V| + |E|)$
2. Insert nodes of in-degree 0 in a FIFO queue (Q)  $O(|V|)$
3. While Q is not empty
4.      $u = Dequeue(Q)$
5.     print u   // Insert in the Sorted List
6.     for each  $v \in G.adj[u]$
7.          $inDegree[v]--$
8.         if  $inDegree[v] == 0$
9.              $Enqueue(Q, v)$

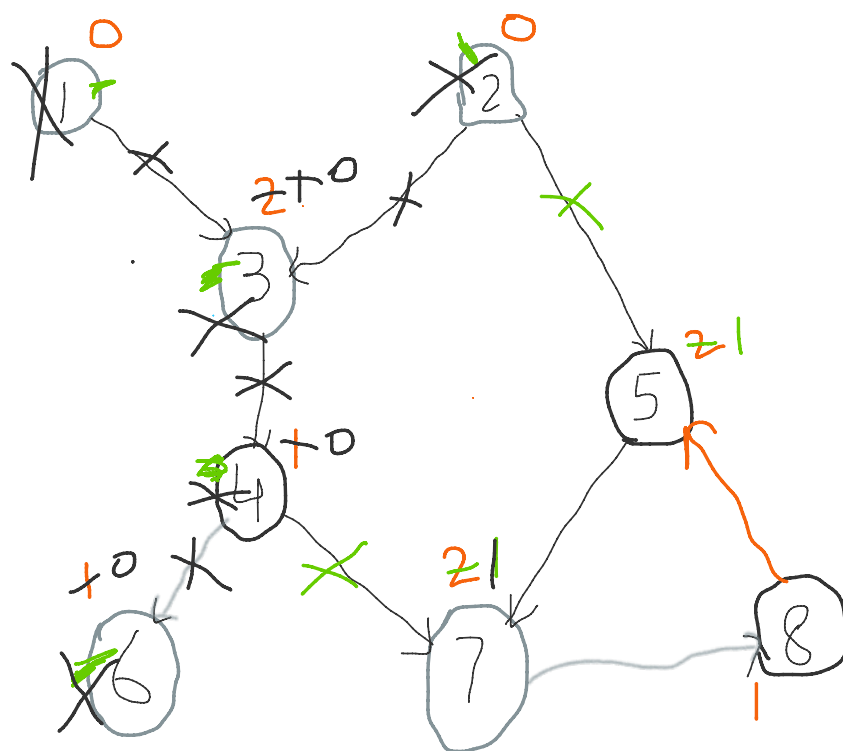
$O(|V| + |E|)$

Time Complexity of Topological Sort using Kahn's algorithm is  $O(|V| + |E|)$  i. e Linear Time Complexity

```

2  1  5  3  4  6  7  8
1  2  3  5  4  7  8  6

```



1 2 3 4

(4,6)  
u v

1 2 3 4 5 6 7 8