Handout: Topologial Sort
CPE 202

Here is an algorithm for performing a topological sort:

1. Build an adjacency list for all of the vertices ***and include*** each vertex's *in degree* (number of incoming edges) as well as the specific vertices adjacent to it.
2. Push all vertices with an *in degree* of zero on to a stack.
3. While the stack is not empty:
    1. Pop and output a vertex.
    2. Reduce the *in degree* of all vertices that were adjacent to the just-popped vertex.
    3. Push all new vertices with an *in degree* of zero.

Given the following edge file and Adjacency List for a directed acyclic graph (DAG):

```
v1 v2
v1 v3
v2 v4
v2 v5
v1 v4
v3 v6
v5 v4
v4 v3
v4 v6
v7 v6
v4 v7
v5 v7
```

V	In Degree	Adjacent Vertices
v1	**0**	**v2, v3, v4**
v2	**1**	**v4, v5**
v3	**2**	**v6**
v4	**3**	**v3, v6, v7**
v5	**1**	**v4, v7**
v6	**3**	
v7	**2**	**v6**

Here is a table that represents the behavior of the provided algorithm:

	In-degree before pop number						
	1	**2**	**3**	**4**	**5**	**6**	**7**
v1	0	-	-	-	-	-	-
v2	1	0	-	-	-	-	-
v3	2	1	1	**1**	**0**	-	-
v4	3	2	1	**0**	-	-	-
v5	1	1	0	-	-	-	-
v6	3	3	3	**3**	**2**	**1**	**0**
v7	2	2	2	**1**	**0**	-	-
push	v1	v2	v5	**v4**	**v3, v7**		**v6**
pop	v1	v2	**v5**	**v4**	**v7**	**v3**	**v6**