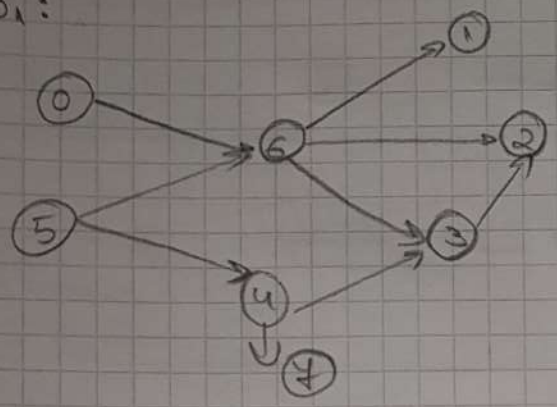
Topological sorting - lab 4
using predecessor counting alg.

G_1:



	x, y	count: dict	q: queue	sorted list
init		0 1 2 3 4 5 6 7 0 1 2 2 1 0 2 1	0 5	[ ]
init 1	x=0 y=6	0 1 2 3 4 5 6 7 0 1 2 2 1 0 1 1	5	[0]
init 2	x=5 y=6 y=4	0 1 2 3 4 5 6 7 0 1 2 2 0 0 0 1	6 6 4	[0,5]
init 3	x=6 y=1 y=3 y=2	0 1 2 3 4 5 6 7 0 0 1 1 0 0 0 1	4 1 - 4 - - v -	[0,5,6]
init 4	x=4 y=3 y=7	0 1 2 3 4 5 6 7 0 0 1 0 0 0 0 0	1 3 1 3 7	[0,5,6,4]
init 5	x=1 -	- v -	3 7	[0,5,6,4,1]

	x, y	count : dict	g : queue	sorted : list
unit 6	x = 3 y = 2	0 1 2 3 4 5 6 7 0 0 0 0 0 0 0 0	7 | 2	[0, 5, 6, 4, 1, 3]
unit 7	x = 7 —	— 4 —	2	[0, 5, 6, 4, 1, 3, 7]
unit 8	x = 2 —	— 1 —		[0, 5, 6, 4, 1, 3, 2] size of (sorted) = 7 G is a DAG

Activity	Durations	Prerequisites
0	1	—
5	2	—
6	5	0, 5
4	1	5
1	2	6
3	2	4, 6
2	1	3, 6
7	2	4