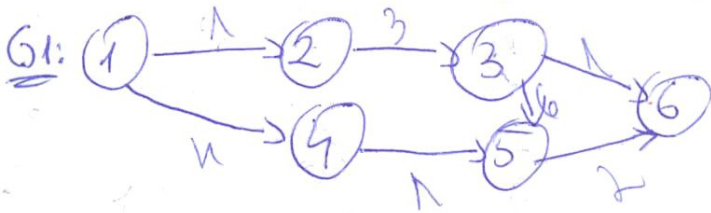


# Topological sorting using predecessor counting algorithm



topological sorting orders:

1, 2, 3, 4, 5, 6

1, 2, 4, 3, 5, 6

1, 4, 2, 3, 5, 6

	X, y	count: dictionary	q: queue	sorted: list
initialization		<div>1 2 3 4 5 6</div> <div>0 1 1 1 1 2</div>	← 1 ←	[]
iteration 1	X=1 y=2 y=4	<div>1 2 3 4 5 6</div> <div>0 0 1 0 1 2</div>	← ← ← 2 4 ←	[1]
iteration 2	X=2 y=3	<div>1 2 3 4 5 6</div> <div>0 0 0 0 1 2</div>	← 4 ← ← 4 3 ←	[1, 2]
iteration 3	X=4 y=5	<div>1 2 3 4 5 6</div> <div>0 0 0 0 1 2</div>	← 3 ←	[1, 2, 4]
iteration 4	X=3 y=5 y=6	<div>1 2 3 4 5 6</div> <div>0 0 0 0 0 1</div>	← ← ← 5 ←	[1, 2, 4, 3]
iteration 5	X=5 y=6	<div>1 2 3 4 5 6</div> <div>0 0 0 0 0 0</div>	← ← ← 6 ←	[1, 2, 4, 3, 5]
iteration 6	X=6		← ← stop	[1, 2, 4, 3, 5, 6] G1 is a DAG size of (sorted) = 6



G2 is not a DAG, has a cycle, there is not topological order of the vertices.

	X, y	count	q	sorted
initializ.		<div>1 2 3</div> <div>0 2 1</div>	← 1 ←	[ ]
iteration 1	X=1 y=2	<div>1 2 3</div> <div>0 1 1</div>	← ←	[1]
			stop	size of (sorted) = 1 ≠ 3 ⇒ ⇒ G2 is not a DAG.