1. A *graph* $G=(V,E)$ consists of a set of *vertices*, *V*, and a set of *edges*, *E*.
2. Each edge is a pair (v,w) where $v,w \in V$.
3. Edges are sometimes referred to as arcs.
4. Directed edges are represented as ordered pairs (v,w). The edge is directed from vertex *v* to vertex *w*. Vertex *v* is the origin and vertex *w* is the destination.

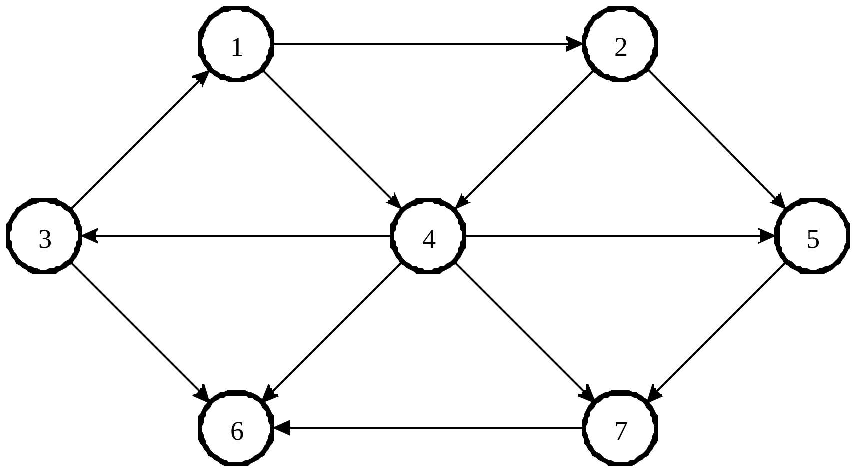

Figure 1. Graph G.

V={1,2,3,4,5,6,7}
E={(1,2),(1,4),(2,4),(2,5),(3,1),(3,6),(4,3),(4,5),(4,6),(4,7),(5,7),(7,6)}

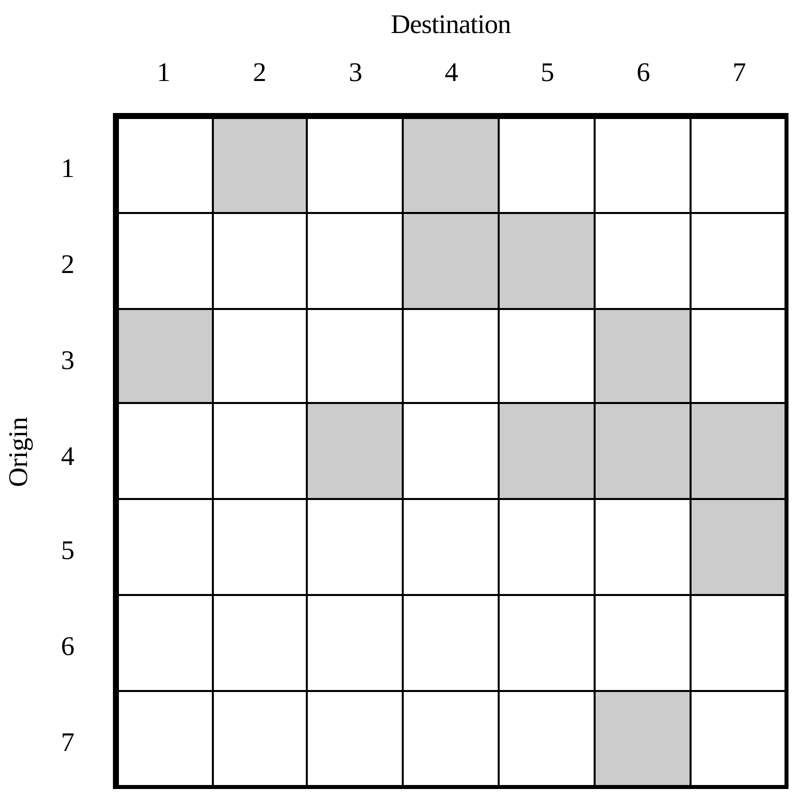

Figure 2. Matrix representation of edges.

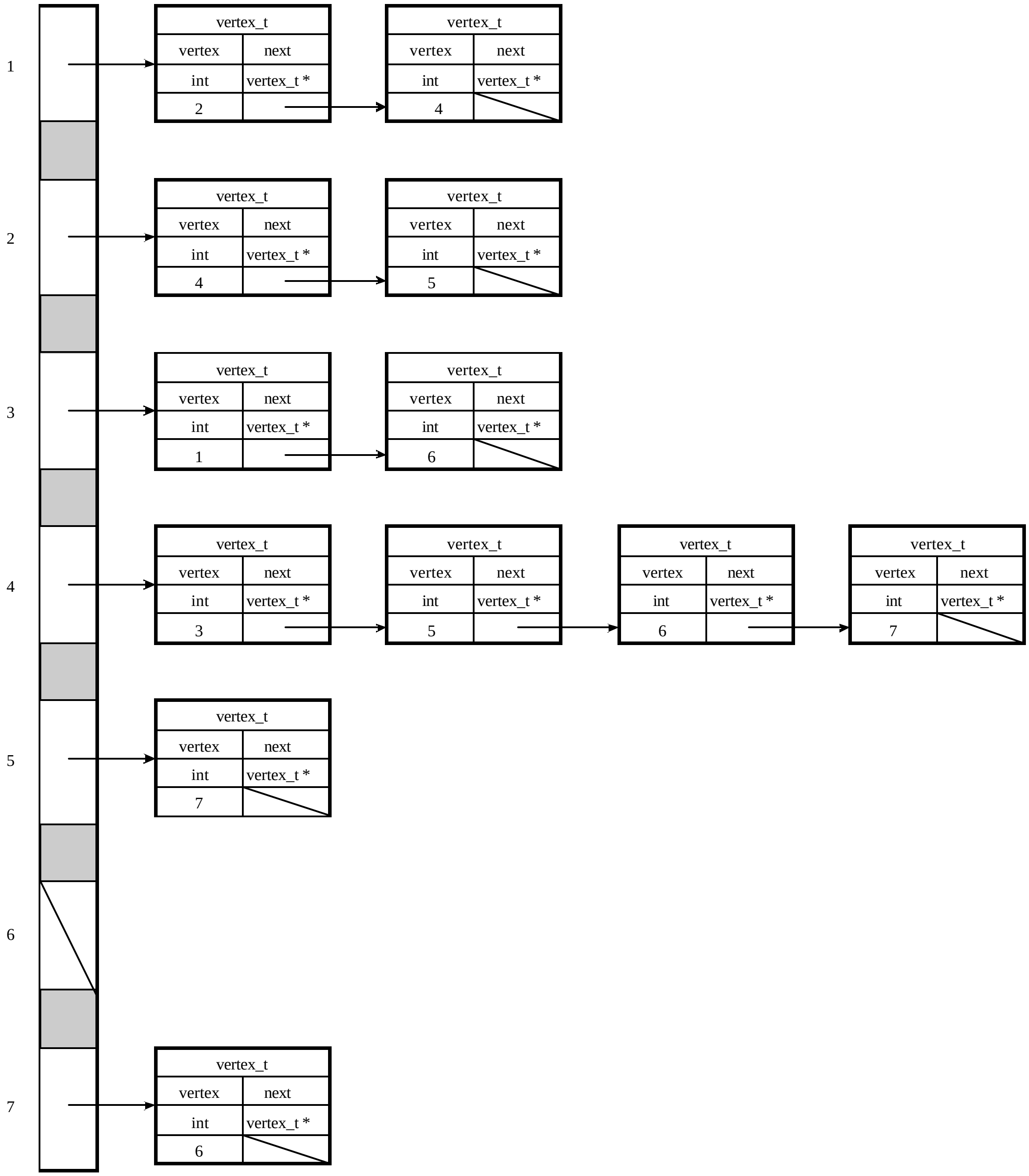

Figure 3. List representation of edges.

Topological Sort

A *topological sort* is an ordering of vertices in a directed acyclic graph, such that if there is a path from v_i to v_j, then v_j appears *after* v_i in the ordering. In the graph in Figure 1, 1, 2, 5, 4, 3, 7, 6 and 1, 2, 5, 4, 7, 3, 6 are both topological orderings.

To perform a topological sort we must define the *indegree* of a vertex. The *indegree* of a vertex *v* is the number of edges (*u*,*v*). The *indegree* of a vertex *v* is the number edges directed to v.

Algorithm:

1. Create queue ***Q*** having an entry for each vertex in the graph.
2. Enqueue all vertexes having an indegree of zero. Enqueue all vertexes that have only outgoing edges.
3. Loop while queue ***Q*** is not empty.
   - 3.1. Dequeue vertex ***V*** from queue ***Q***
   - 3.2. For each vertex ***W*** adjacent to ***V***
     - 3.2.1. Decrement the indegree of vertex ***W***
     - 3.2.2. Enqueue vertex W if its *indegree* is zero

	Indegree Before Dequeue #						
Vertex	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	0	0	0	0	0	0	0
2	1	*0*	0	0	0	0	0
3	2	*1*	1	1	*0*	0	0
4	3	*2*	*1*	*0*	0	0	0
5	1	1	*0*	0	0	0	0
6	3	3	3	3	*2*	*1*	0
7	2	2	1	*1*	*0*	0	*0*
Q	{1}	{2}	{5}	{4}	{3,7}	{7}	{6}
Dequeue	1	2	5	4	3	7	6
Adjacent	2,3,4	4,5	4,7	3,6,7	6	6	

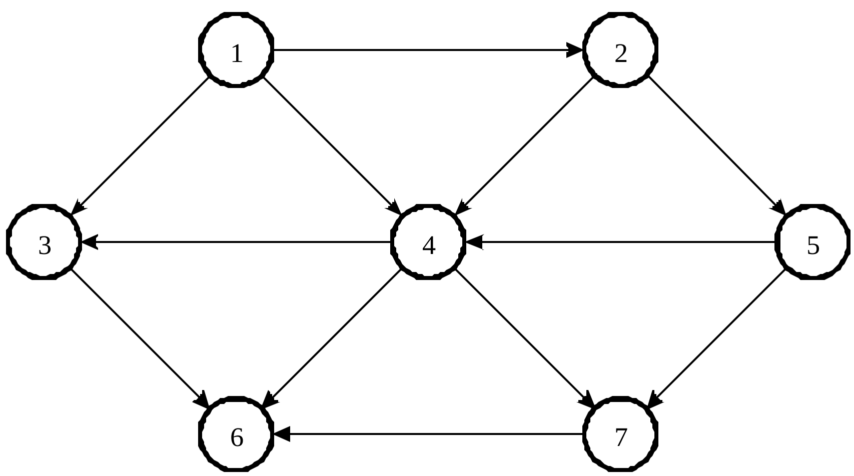

Figure 4. Graph for Topological Sort