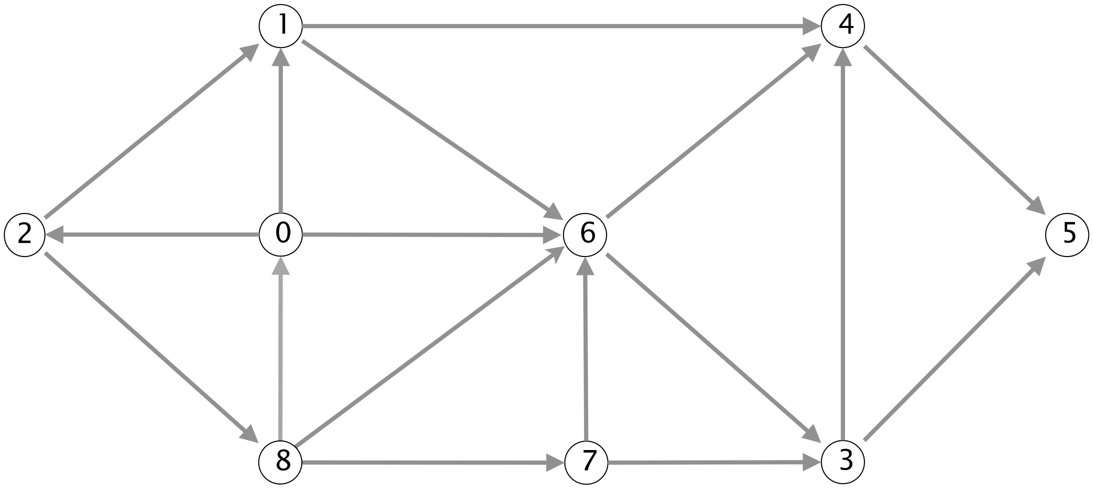
3. **Graph search. (8 points)**

Consider the following digraph. Assume the adjacency lists are in sorted order: for example, when iterating through the edges pointing from 0, consider the edge 0 → 1 before 0 → 2 or 0 → 6.



(a) Run depth-first search on the digraph, starting from vertex 0. List the vertices in *reverse postorder*.

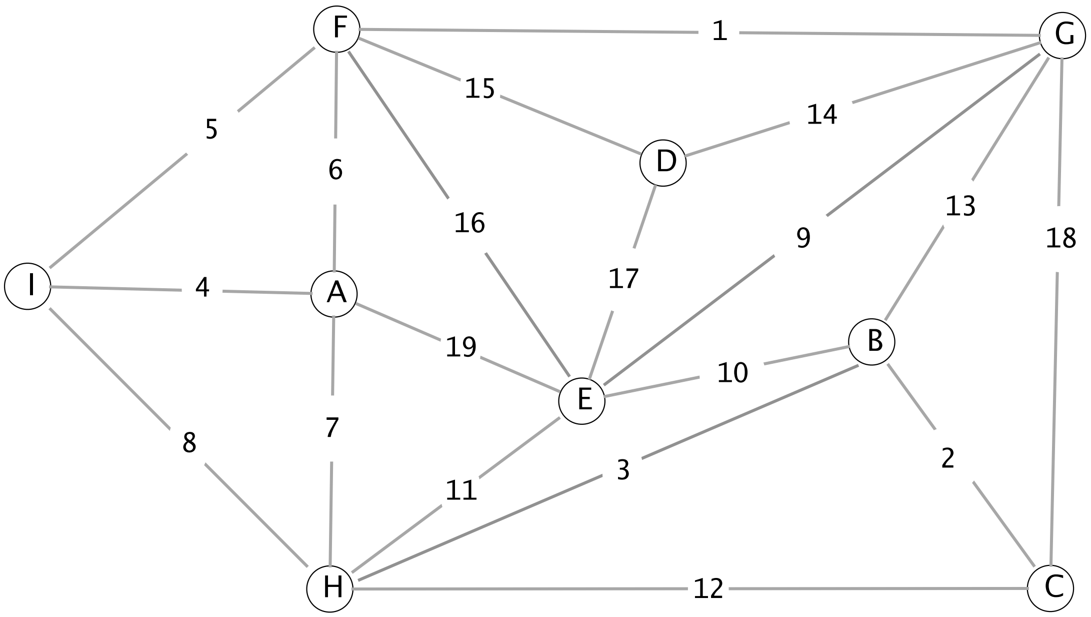
```
 0
---    ---    ---    ---    ---    ---    ---    ---    ---
```

(b) Run breadth-first search on the digraph, starting from vertex 0. List the vertices in the order in which they are dequeued from the FIFO queue.

```
 0
---    ---    ---    ---    ---    ---    ---    ---    ---
```

4. **Minimum spanning trees. (8 points)**

   Consider the following edge-weighted graph with 9 vertices and 19 edges. Note that the edge weights are distinct integers between 1 and 19.



   (a) Complete the sequence of edges in the MST in the order that *Kruskal's algorithm* includes them (by specifying their edge weights).

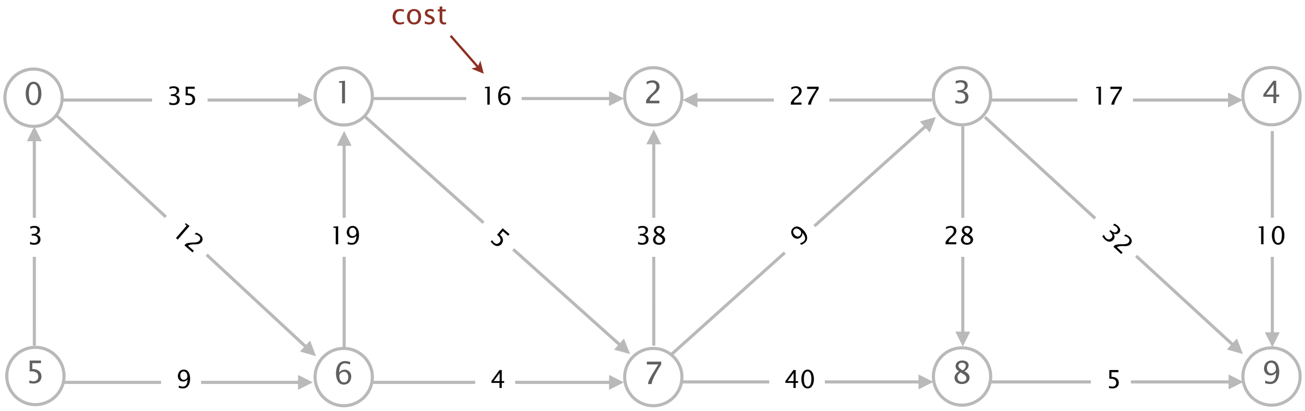
```
 1
---- ---- ---- ---- ---- ---- ---- ----
```

   (b) Complete the sequence of edges in the MST in the order that *Prim's algorithm* includes them (by specifying their edge weights), starting from vertex *A*.

```
 4
---- ---- ---- ---- ---- ---- ---- ----
```

4. **Shortest paths. (6 points)**

Suppose that you are running Dijkstra's algorithm on the edge-weighted digraph below, starting from some vertex s (not necessarily 0).



The table below gives the `edgeTo[]` and `distTo[]` values immediately after vertex 7 has been deleted from the priority queue and relaxed.

v	distTo[]	edgeTo[]
0	3.0	5 → 0
1	28.0	6 → 1
2	51.0	7 → 2
3	22.0	7 → 3
4	∞	*null*
5	0.0	*null*
6	9.0	5 → 6
7	13.0	6 → 7
8	53.0	7 → 8
9	∞	*null*

(a) Give the order in which the first 4 vertices were deleted from the priority queue and relaxed.

			7

(b) Which is the *next* vertex after 7 to be deleted from the priority queue and relaxed?

0 1 2 3 4 5 6 7 8 9

(c) In the table below, fill in those entries (*and only those entries*) in the `edgeTo[]` and `distTo[]` arrays that change (from the corresponding entries on the facing page) immediately after the next vertex after 7 is deleted from the priority queue and relaxed.

`v`	`distTo[]`	`edgeTo[]`
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		