



Problem 1. Perform a breadth-first search of the graph starting from vertex A. Give the number of steps to reach *every* other vertex. Additionally, give the order in which the vertices are *first* witnessed; that is, give the order in which they first enter the queue (and not necessarily the order in which they are explored).

The following represents the queue as BFS is performed on the graph

1. queue: B, C
2. queue: C, E, D
3. queue: E, D
4. queue: D, F
5. queue: F

The number of steps to reach every other vertex is 5. The order in which each vertex is added to the queue is B, C, E, D, F.

Problem 2. Use Dijkstra's algorithm on this graph starting from vertex A. Give the cost of the least-cost path to *every* other vertex. Additionally, give the order in which the vertices are *first* witnessed; that is, give the order in which they first enter the queue (and not necessarily the order in which they are explored).

The following represents the priority queue as SSSP is performed.

1. queue: 5:B, 7:C
2. queue: 6:C, 7:C, 9:D, 13:E
3. queue: 7:C, 8:D, 9:D, 13:E
4. queue: 8:D, 9:D, 13:E
5. queue: 9:D, 13:E, 16:F
6. queue: 13:E, 16:F

7. queue: 16:F, 19:F

8. queue: 19:F

9. queue: empty

The least cost to reach each node is the following: B:5, C:6, D:8, E:13, F:16

The order in which the vertex are first witnessed is B, C, D, E, F.

Problem 3. Give two valid topological sorts of this graph.

Sort 1	Sort 2
A	A
B	B
E	C
C	D
D	E
F	F