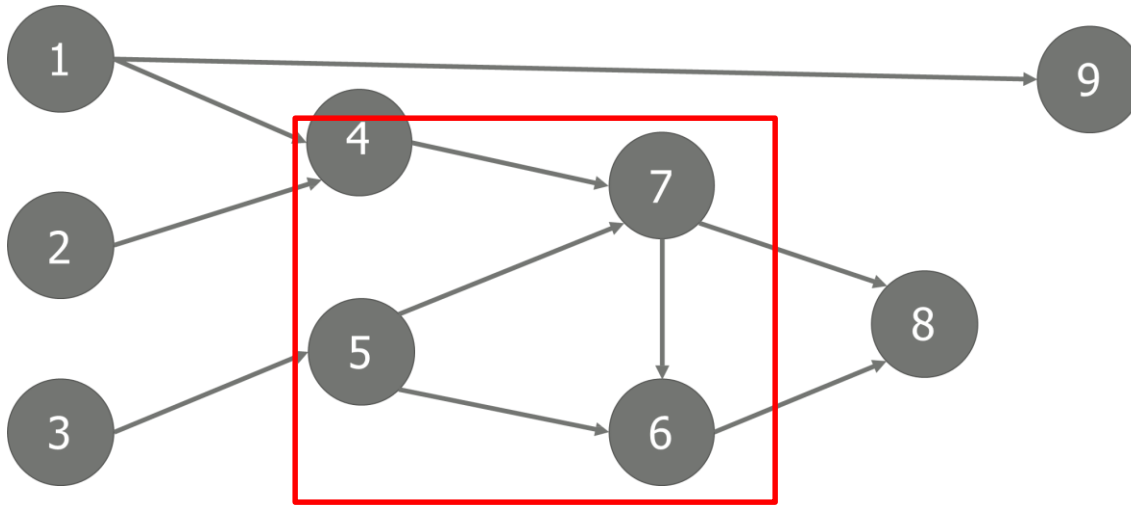


DAG – 위상 정렬

Directed Acyclic Graph

- 사이클이 없는 방향이 있는 그래프를 DAG(Directed Acyclic Graph) 라고 한다.
- DAG에서는 위상 정렬 문제를 모델링 할 수 있다.

위상 정렬 : 어떤 일을 하는 순서를 찾는 알고리즘

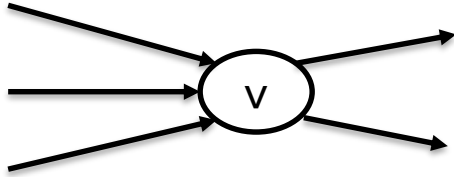


- 1,2,3,4,5,6,7,8,9 차례대로 노드를 방문하지 못한다.
6을 방문하기 전에는 5와 7이 먼저 방문 된 후에 6이 방문 되어야 하기 때문이다.
- 1,2,3,4,5,7,6,8,9 는 위상 정렬 답 중 하나가 될 수 있다.

DAG – 위상 정렬

Directed Acyclic Graph

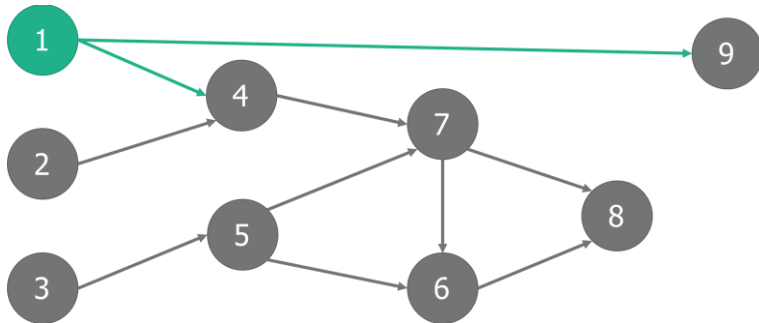
- 위상 정렬의 방문 순서를 정하기 위해서 indgree / outdegree를 따로 저장한다.



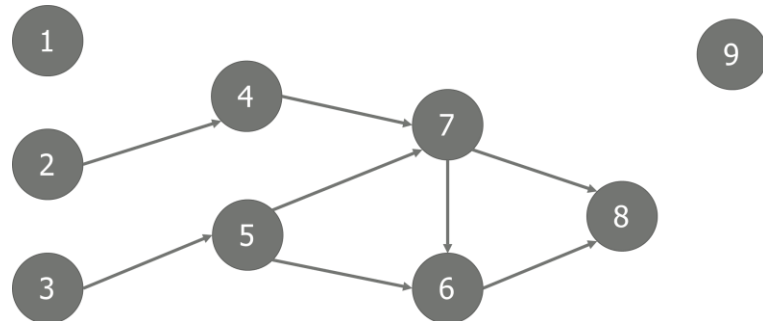
- indgree : 3 / outdegree : 2
- 위상 정렬 시 노드가 방문되기 위해서 indgree의 개수 만큼 방문 되어야 한다.
→ 방문이 되는 경우 indgree의 개수를 하나 씩 줄여 indgree가 0이되면 큐에 저장한다.

- indgree가 0인 1, 2, 3 노드를 큐에 넣는다.

- 큐 : 1, 2, 3 / 순서 :



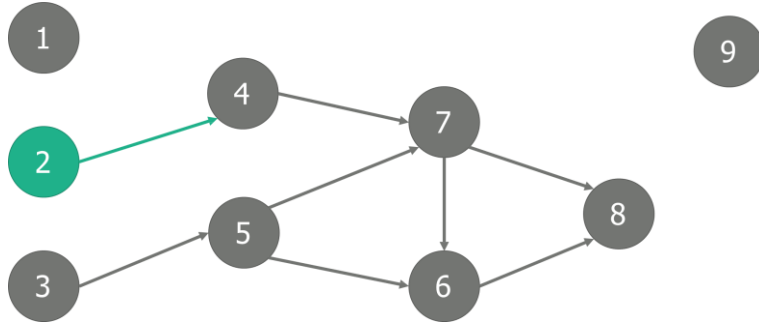
- 큐 : 2, 3, 9 / 순서 : 1



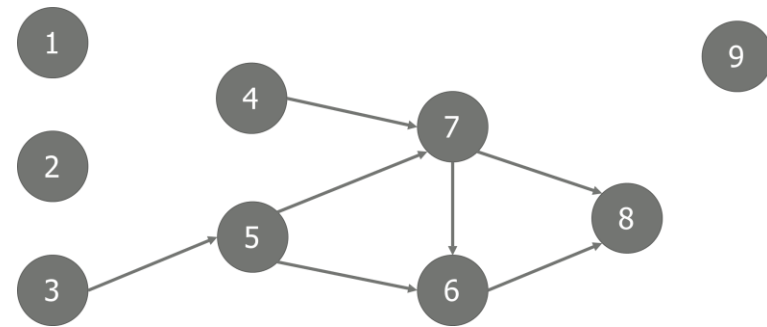
DAG – 위상 정렬

Directed Acyclic Graph

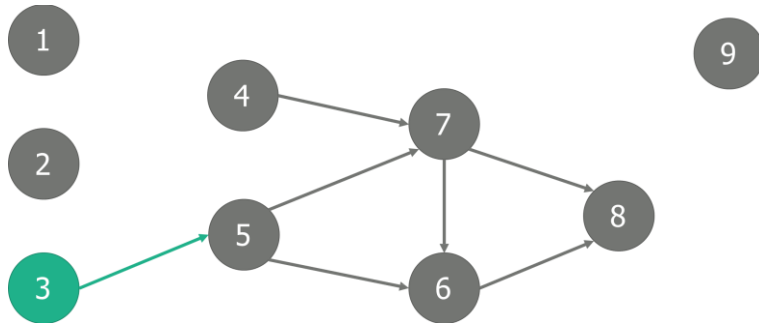
- 큐 : 2, 3, 9 / 순서 : 1



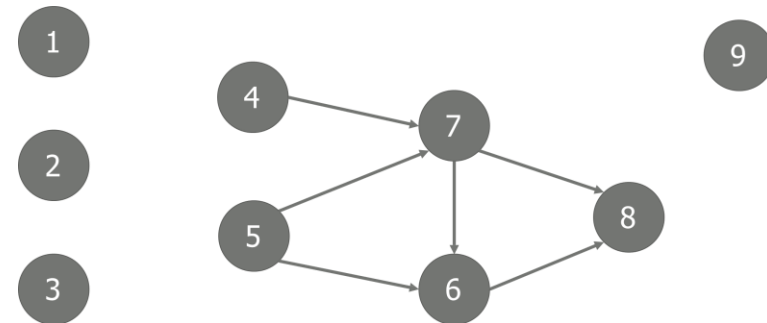
- 큐 : 3, 9, 4 / 순서 : 1, 2



- 큐 : 3, 9, 4 / 순서 : 1, 2



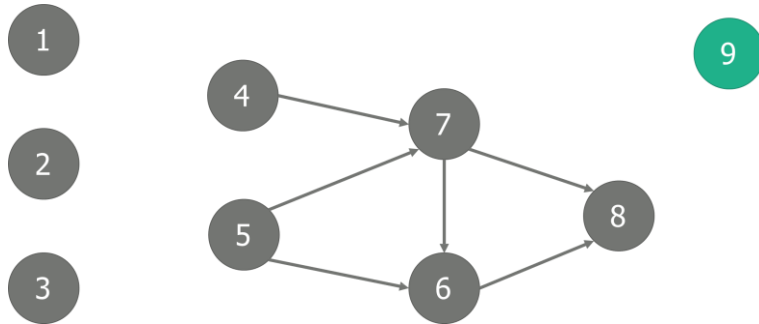
- 큐 : 9, 4, 5 / 순서 : 1, 2, 3



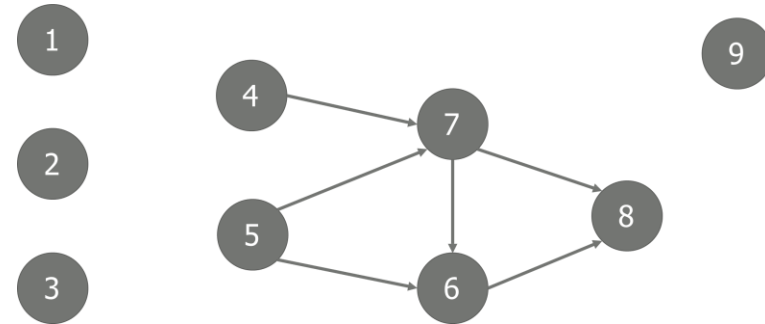
DAG – 위상 정렬

Directed Acyclic Graph

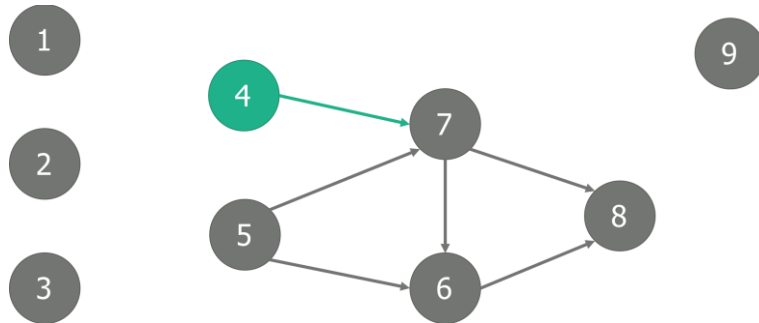
- 큐 : 9, 4, 5 / 순서 : 1, 2, 3



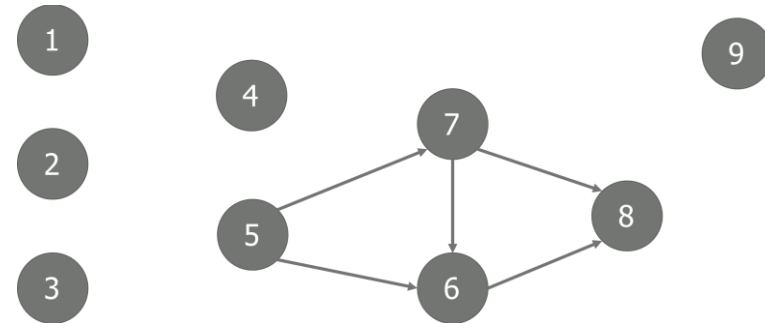
- 큐 : 4, 5 / 순서 : 1, 2, 3, 9



- 큐 : 4, 5 / 순서 : 1, 2, 3, 9



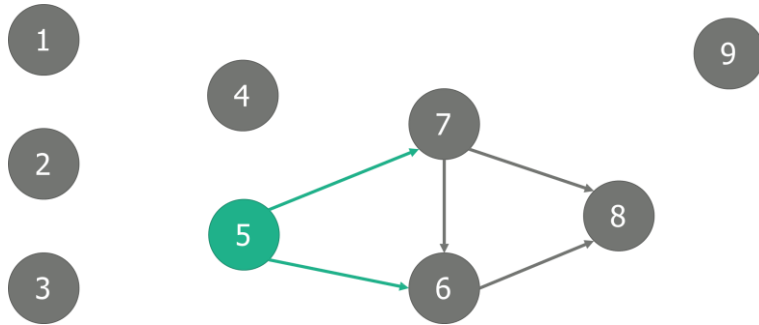
- 큐 : 5 / 순서 : 1, 2, 3, 9, 4



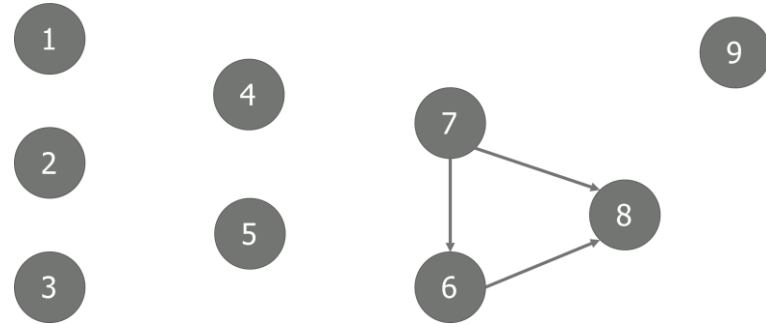
DAG – 위상 정렬

Directed Acyclic Graph

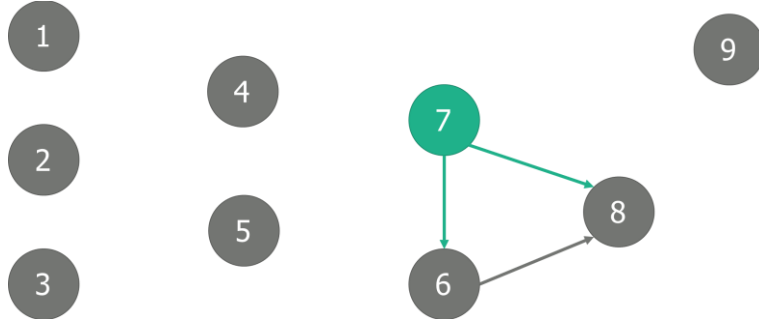
- 큐 : 5 / 순서 : 1, 2, 3, 9, 4



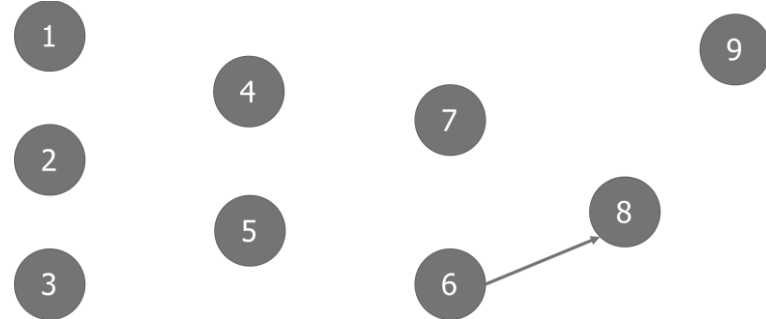
- 큐 : 7 / 순서 : 1, 2, 3, 9, 4, 5



- 큐 : 7 / 순서 : 1, 2, 3, 9, 4, 5



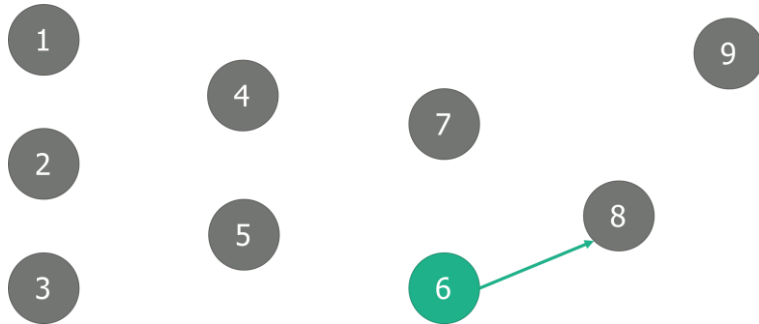
- 큐 : 6 / 순서 : 1, 2, 3, 9, 4, 5, 7



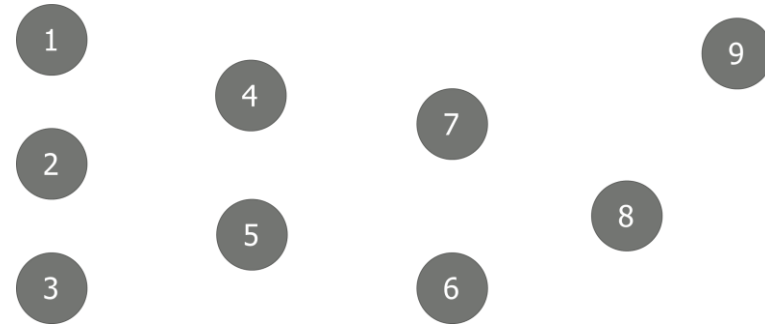
DAG – 위상 정렬

[Directed Acyclic Graph](#)

- 큐 : 6 / 순서 : 1, 2, 3, 9, 4, 5, 7



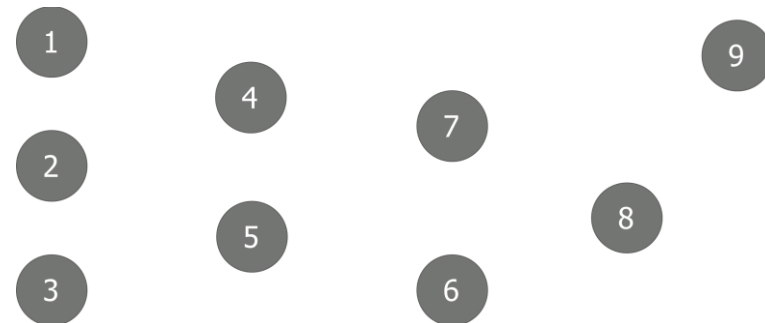
- 큐 : 8 / 순서 : 1, 2, 3, 9, 4, 5, 7, 6



- 큐 : 8 / 순서 : 1, 2, 3, 9, 4, 5, 7, 6



- 큐 : / 순서 : 1, 2, 3, 9, 4, 5, 7, 6, 8



DAG – 위상 정렬

Directed Acyclic Graph

- 큐 대신에 우선순위 큐를 이용하여 방문 순서의 기준을 정하면 오직 하나의 답만 나올 수 있다.
- 우선순위 큐 : 9, 4, 5 / 순서 : 1, 2, 3
- 우선순위 큐(minheap)
: 9, 4, 5 / 순서 : 1, 2, 3

