

## 1. 구현 아이디어

주어진 graph의 data를 가지고 adjacency list로 구현하였다. 이후 DFS 알고리즘을 사용하여 graph의 vertex를 출력하고 topology sort의 결과를 출력한다. topology sort는 주어진 graph가 directed graph이면서 DAG이기 때문에 각 vertex의 나가는 edge 배열인 outDegree 배열을 사용하여 구현하였다.

## 2. 구현

---

**Algorithm 1:** topology sort

---

```

1  $Q \leftarrow \emptyset$ 
2 for  $i$  to  $D.length$  do
3   if  $D[i] == 0$  then
4     enqueue( $Q, i$ )
5   end
6 end
7 while  $Q \neq \emptyset$  do
8    $n \leftarrow dequeue(Q)$ 
9   print( $n$ )
10  for  $i$  to  $A[n].length$  do
11     $D[A[n][i]] -= 1$ 
12    if  $D[A[n][i]] == 0$  then
13      enqueue( $Q, D[A[n][i]]$ )
14    end
15  end
16 end

```

---

11번 줄에서 선행자를 삭제하는 방법으로 구현하였다. 예를 들어 (0, 1), (0, 2)이면 0에서 1, 2 vertex로 나가는 edge가 존재한다. 원래  $D[0] = 2$ 에서 1에 도달하면 -1을 하고 2에 도달하면 -1하게 된다. 그러면 0이되어 queue에 추가한다. 이를 반복하면 정렬된 결과를 출력하게 된다.

## 3. 결과

