

Algorithm

- topology sort -

제출일자	2020.11.20
과제 번호	11
이 름	강인한
학 번	201701969

실행 결과

```
Topology Sort Result
cycle
0
1
3
2
```

```
Topology Sort Result
0
1
2
3
4
5
```

풀이과정 : topology sort를 구현하는 방법은 다양합니다. 그중에서 제가 선택한 방법은 dfs방식으로 접근하여 dfs가 끝난 vertex를 역순으로 출력하면 topology sort가 되는 방법을 이용했습니다. 또한 arraylist에 vertex에서 vertex로 향하는 edge를 저장하고 array에서 큰 값을 갖는 vertex에서 작은 값을 갖는 vertex로의 edge가 있다면 cycle임을 이용하여 cycle을 찾아냈습니다. cycle을 찾는 방식으로는 vertex의 개수보다 많이 반복하여 stack에 push되는 vertex를 통해 cycle을 이루는 vertex를 찾을 수 있었습니다.

코드 설명 : dfs처럼 arraylist에 인접한(향하는) vertex의 값을 저장하고, stack에 0을 넣고 0의 값을 갖는 vertex에서 다음 vertex들을 stack에 저장하는 식으로 진행한다. stack에 들어간 vertex중에 top위치에 있는 vertex의 다음 vertex를 stack에 push하는 과정을 계속 반복한다. 그러다 이전 vertex의 값이 이후 vertex의 값보다 크면 이는 역방향 edge이므로 cycle이 존재한다는 것을 알 수 있다.

i) cycle이 아니라면 출력으로 넘어가 방문을 마친 vertex부터 역순으로 출력한다. 이러한 과정을 통해 위상정렬을 출력할 수 있다.

ii) cycle이라면 가장 뒤에 접근 된(stack에 top에 위치한) 한 cycle만큼 출력한다.

느낀 점 : cycle에 해당되는 vertex만 출력하는 게 많이 어려운 것 같습니다.