위상정렬

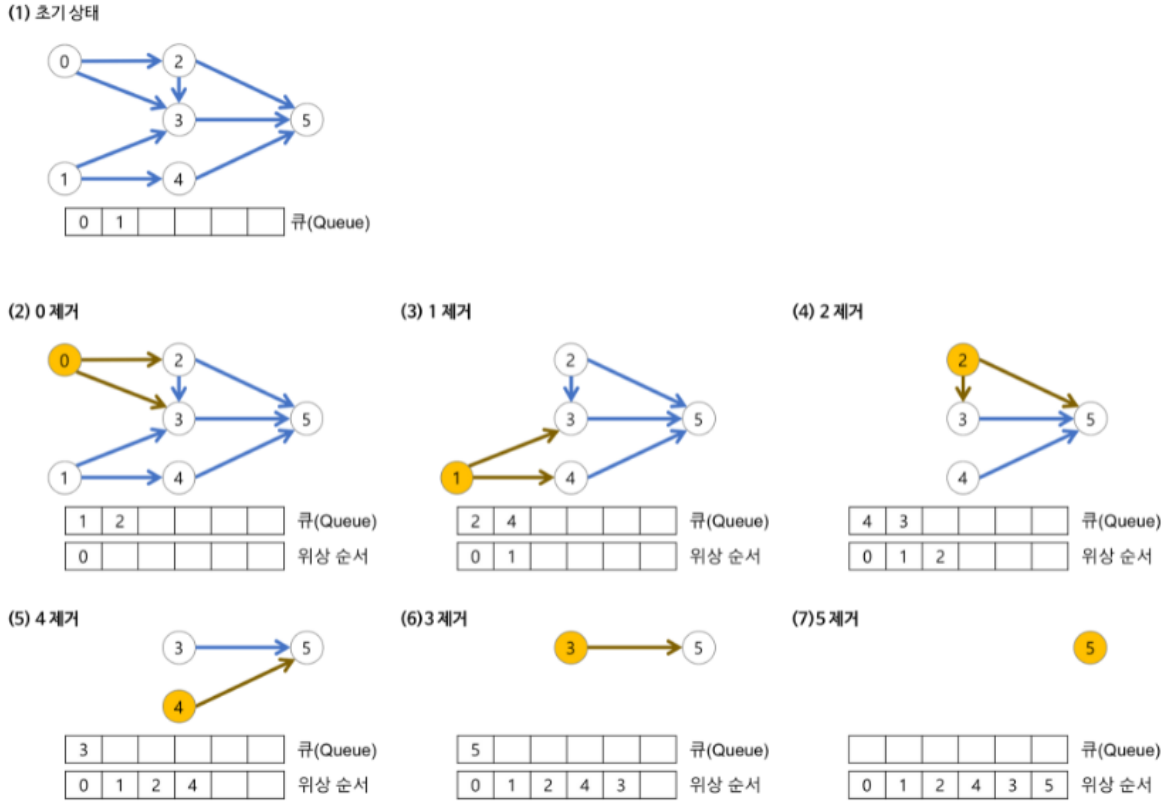
☑ 시간복잡도

- 최선, 최악, 평균 $O(V+E)$

☑ 개요

- **위상 정렬**(topological sorting)은 유향 그래프의 꼭짓점들(vertex)을 변의 방향을 거스르지 않도록 나열하는 것을 의미한다.
- 하나의 방향 그래프에는 여러 위상 정렬이 가능하다.
- 위상 정렬의 과정에서 선택되는 정점의 순서를 위상 순서(Topological Order)라 한다.
- 위상 정렬의 과정에서 그래프에 남아 있는 정점 중에 진입 차수가 0인 정점이 없다면, 위상 정렬 알고리즘은 중단되고 이러한 그래프로 표현된 문제는 실행이 불가능한 문제가 된다.

☑ Preview



✅ Code

- 백준 1766 문제집 풀이

```
import sys
from heapq import heappop, heappush
sys.stdin = open('input.txt')
q = lambda : map(int, sys.stdin.readline().split())

N, M = q()

problem = [[] for _ in range(N+1)] # 문제 번호 연결 정보
indegree = [0 for i in range(N + 1)] # 위상 저장

heap = []
order = []

#Graph Information
for i in range(M):
    A, B = q() # 먼저풀거, 나중에 풀거
    problem[A].append(B) # 자식(나중에 풀 문제 번호) 추가
    indegree[B] += 1 # B의 위상 +1 (선수 문제의 갯수)

#Make First heap
for i in range(1, N + 1):
    if indegree[i] == 0: # 진입 루트가 0인 것(선수 문제가 없는 것)
        heappush(heap, i) # 우선순위 큐(= 힙)에 담기

#Make Topological Sort Loop
while heap:
    temp = heappop(heap)
    order.append(temp)
    for j in problem[temp]: # 하위 문제
        indegree[j] -= 1 # 선수문제를 풀었으니, 하위 문제의 위상 -1
        if indegree[j] == 0: # 0이 된 경우 = 선수문제들을 다 푼 경우
            heappush(heap, j)

for i in order:
    print(i, end=' ')
```