

위상정렬

23.9.25 박송ㅇ

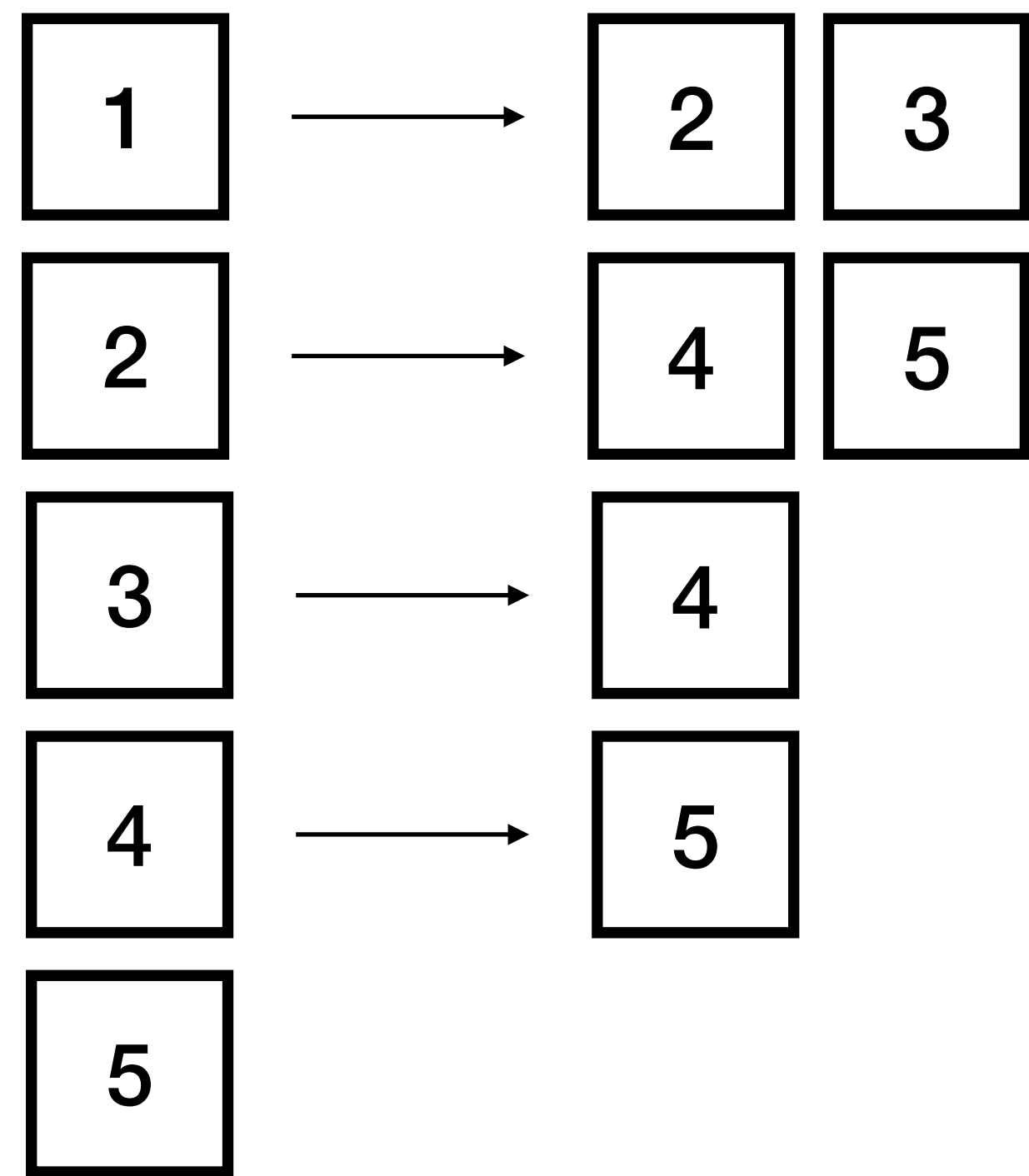
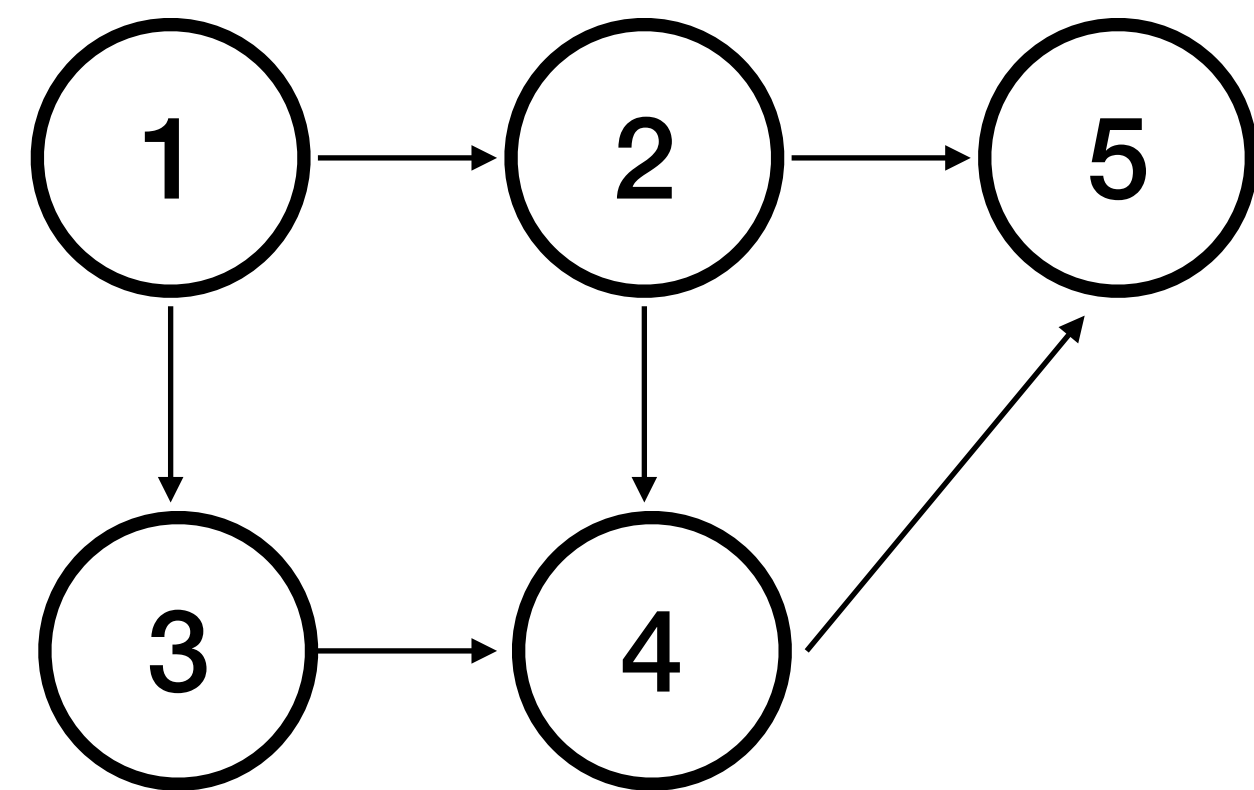
위상정렬 이해하기

- 사이클이 없는 방향 그래프에서 노드 순서를 찾는 알고리즘
- 노드 간의 순서를 결정
- * **사이클이 없어야 한다.**
- 시간 복잡도 $O(V+E)$
- 위상 정렬에서는 유일한 값으로 정렬되지 않는다.
- 사이클이 존재하면 노드 간의 순서를 명확하게 정의할 수 없으므로 위상 정렬 적용 불가능!

위상 정렬 이해하기

진입 차수

진입차수란 자기 자신을 가리키는 에지의 개수



진입 차수 리스트 D[N]

1	1	1	1	1
0	1	1	2	2

위상 정렬 이해하기

진입 차수

진입 차수 리스트 D[N]

1	2	3	4	5
0	1	1	2	2

진입 차수가 0인 노드 선택

위상 정렬 리스트

1

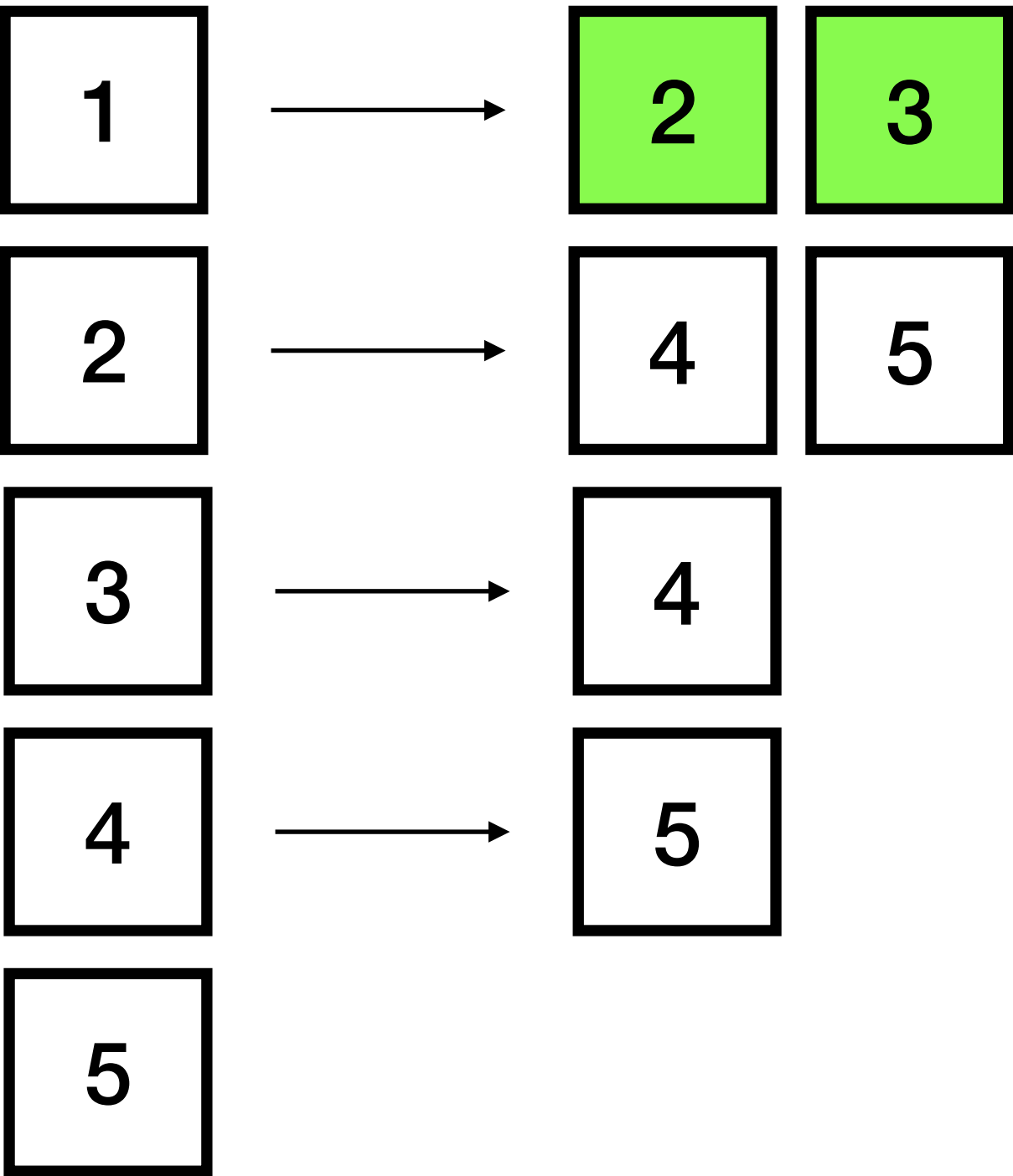
위상 정렬 리스트에 해당 노드 저장

인접 리스트에서 해당 선택된 노드가 가리키는
노드들의 진입 차수를 1씩 빼주기

진입 차수 리스트 D[N]

1	2	3	4	5
0	0	0	2	2

연결 리스트



위상 정렬 이해하기

진입 차수

진입 차수 리스트 D[N]

1	2	3	4	5
0	0	0	2	2

위상 정렬 리스트

1	2	3	4	5
---	---	---	---	---

같은 방식으로 진입 차수가 0인 노드를 골라 위상 정렬 리스트에 넣어주기
해당 노드가 가리키는 노드들의 진입 차수를 1씩 빼주기

이 과정을 모든 노드가 정렬될때까지 반복하기

이때 진입차수가 0인 노드가 또 있을 수 있으므로
위상정렬의 정렬은 항상 일정하지 않음.

문제 53 - 줄 세우기

N명의 학생들을 키 순서대로 줄을 세우기 M번 비교
직접 키를 잴 수 없어 두 학생의 키를 비교하는 방법을 사용한다.

예제 입력

4 2 #N, #M

4 2 #학생번호

3 1

예제 출력

3 4 1 2

학생 -> 노드
키 순서 비교 데이터 -> 에지

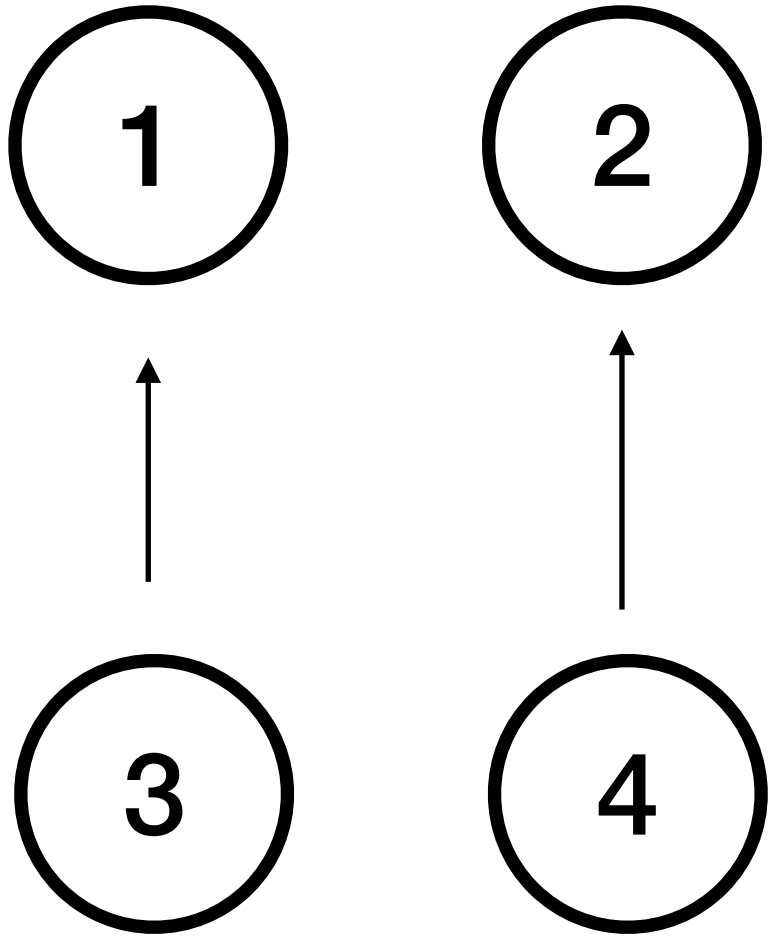
문제 53 - 줄 세우기

예제 입력

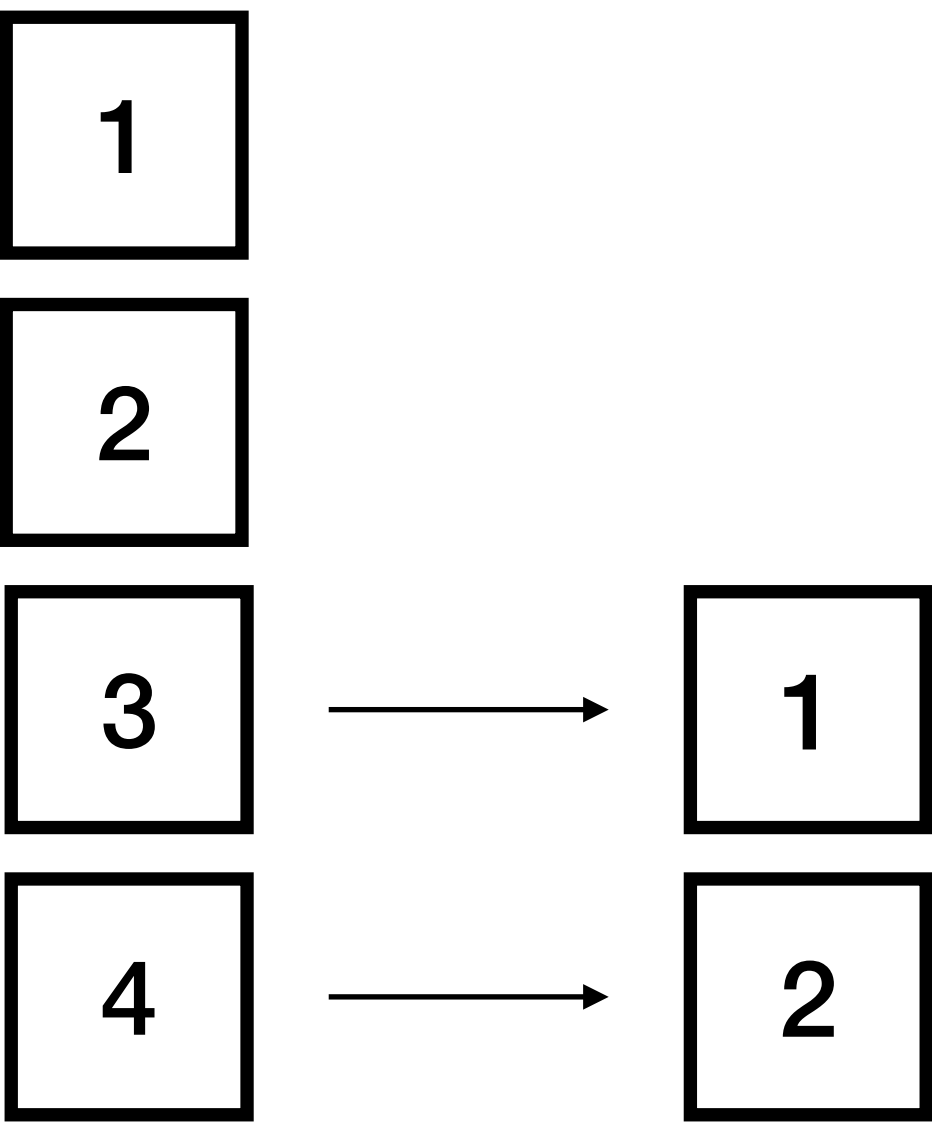
4 2 #N, #M

4 2 #학생번호

3 1



인접 리스트



진입 차수 리스트 D[N]

1	2	3	4
1	1	0	0

문제 53 - 줄 세우기

진입 차수 리스트 D[N]

1	2	3	4
1	1	0	0

1	2	3	4
0	1	0	0

1	2	3	4
1	0	0	0

1	2	3	4
1	1	0	0

위상 정렬 리스트

3

3	4
---	---

3	4	1
---	---	---

3	4	1	2
---	---	---	---

변경된 진입 차수 리스트 D[N]

1	2	3	4
0	1	0	0

1	2	3	4
1	0	0	0

1	2	3	4
1	1	0	0

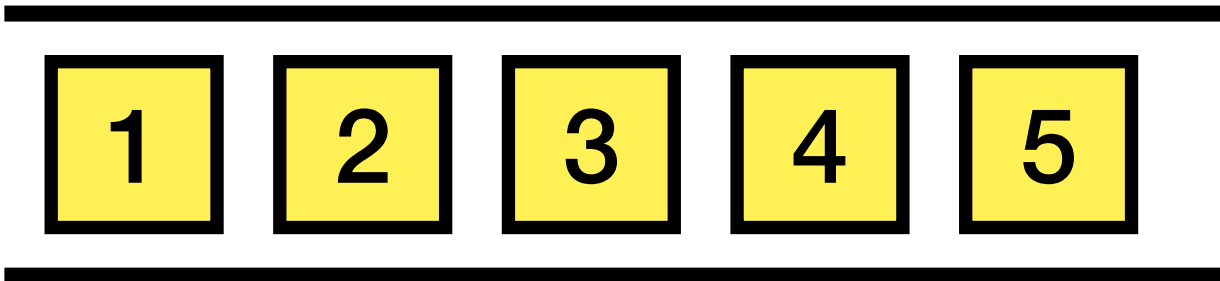
문제 53 - 줄 세우기

```
N,M = map(int,input().split())
A = [[] for _ in range(N+1)]
indegree = [0] * (N+1)

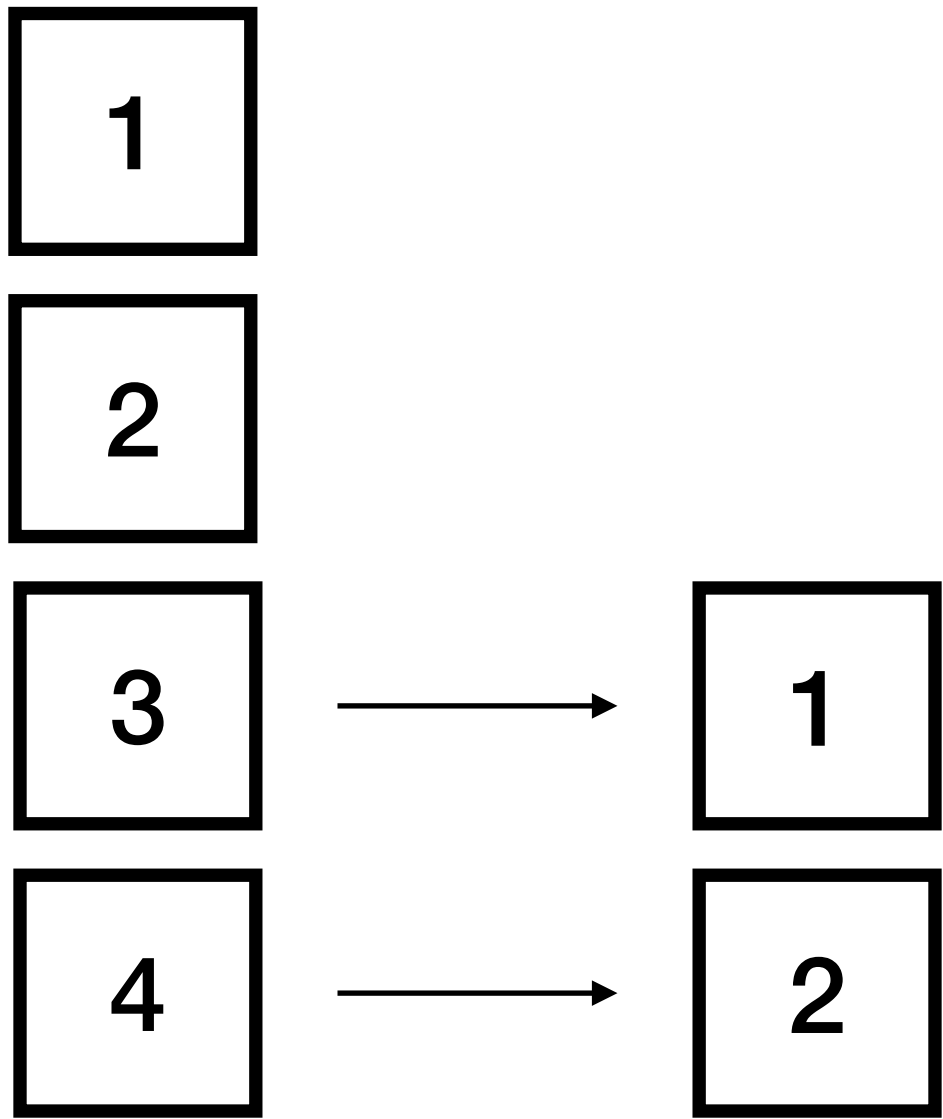
for _ in range(M):
    S,E = map(int,input().split())
    A[S].append(E)
    indegree[E] += 1

queue = deque()
```

위상 정렬 queue



인접 리스트 A



진입 차수 리스트

indegree

1	2	3	4
1	1	0	0

문제 53 - 줄 세우기

```
while queue:
    now = queue.popleft()
    print(now, end=' ')
    for next in A[now]:
        indegree[next] -= 1
        if indegree[next] == 0:
            queue.append(next)
```

queue에서 제일 왼쪽에 있는 값을 꺼내기

그 값을 출력

인접 리스트에서 해당 노드에 연결된 노드들을 탐색하면서

그 노드들의 진입 차수를 1씩 빼준다.

만약 진입 차수를 1 뺐을 때 그 값이 0이라면 queue에 넣어준다.

이런식으로 반복하면서 queue가 빌 때까지 반복한다.