위상 정렬

lhd

위상정렬

- 참조 사이트

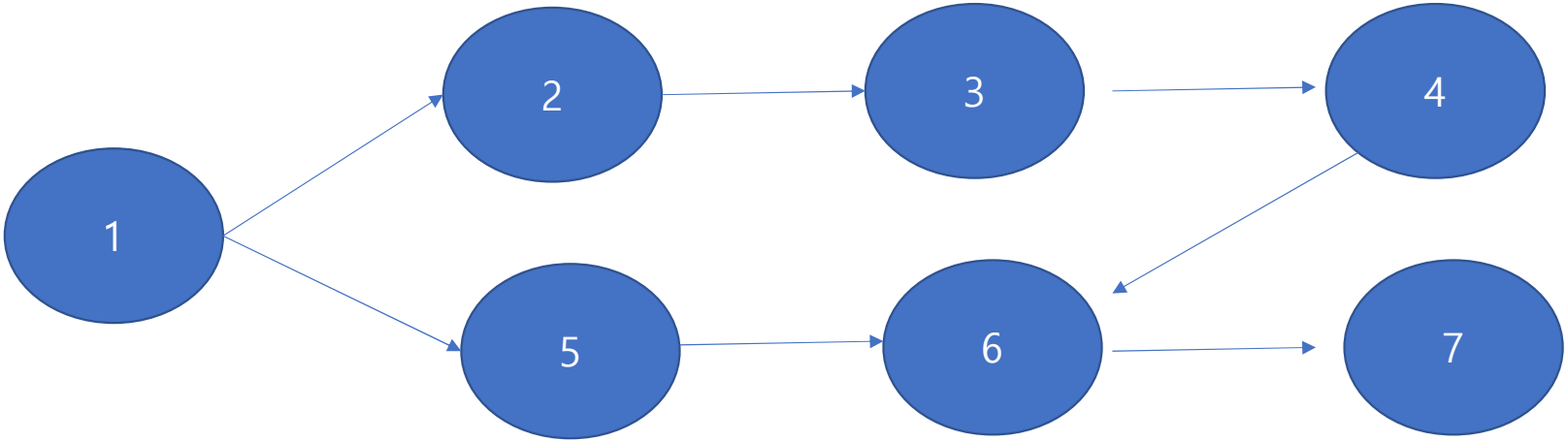
위상정렬

- 사이클이 발생하면 위상정렬을 수행할 수 없다.
- 시작점이 존재해야 만 발생한다.
- Stack
- Queue -> 더 많이 사용
- 집입차수는 특정한 노드가 주어졌을 때 다른 노드가 들어노는 개수를 의미한다.

위상정렬

- 진입차수가 0인 정점을 큐에 삽입합니다.
- 큐에서 원소를 꺼내 연결된 모든 간선을 제거합니다.
- 간선 제거 이후에 진입차수가 0이 된 정점을 큐에 삽입합니다.
- 큐가 빌 때 까지 2번 ~ 3번 과정을 반복합니다. 모든 원소를 방문하기 전에 큐가 빈다면 사이클이 존재하는 것이고 , 모든 원소를 방문했다면 큐에서 꺼낸 순서가 위상 정렬의 결과입니다.

위상정렬



a

0	
1	[2, 5]
2	[3]
3	[4]
4	[6]
5	[6]
6	[7]
7	

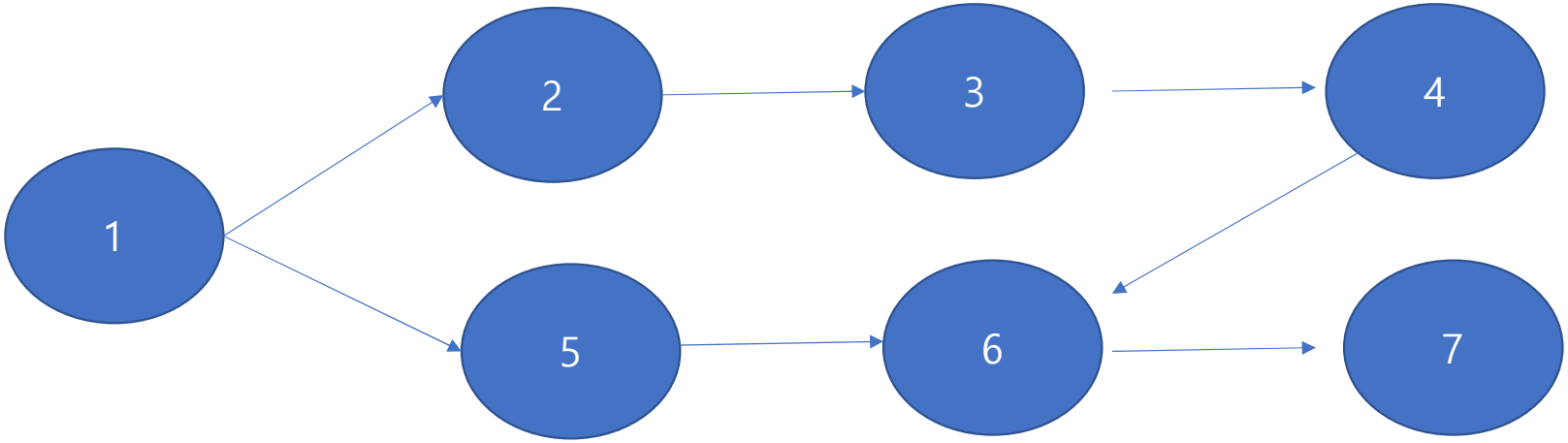
indegree

1	2	3	4	5	6	7
0	1	1	1	1	2	1

위상정렬

```
q = deque()
# 진입 차수가 0인 노드를 큐에 삽입합니다.
for i in range(1,n+1):
    if (inDegree[i] == 0) :
        q.append(i)
print(q)
```

- 진입차수가 0인 것을 다 큐에 넣어준다.
- 진입차수가 0인 것이 하나밖에 없다.



위상정렬

```python
    # 위상 정렬이 완전히 수행되려면 정확히 N개의 노드를 방문합니다.
    for _ in range(1, n+1):
        #n개를 방문하기 전에 큐가 빈다면 사이클이 발생한 것이다
        if not q:
            print("사이클이 발생했습니다.")
            return
        x = q.popleft()
        print(q, x, a[x])
        result.append(x)
        for j in a[x]:
            #y = a[x][j]
            inDegree[j]-= 1
            if inDegree[j] == 0:
                q.append(j)
```

0	
1	[2, 5]
2	[3]
3	[4]
4	[6]
5	[6]
6	[7]
7	

- Q에는 현재 1만 존재
- 1를 꺼내고 result에 넣어주고
- For j -> a[x] -> [2,5]에서 처음에는 indDegree[j] -=1
- indDegree[2]에서 1나 빼면 0으로 변경
- indDegree[2] ==0 이여서 q 에 넣어줌

1	2	3	4	5	6	7
0	1	1	1	1	2	1

1	2	3	4	5	6	7
0	0	1	1	1	2	1

```
deque([1])
deque([]) 1 [2, 5]
deque([5]) 2 [3]
deque([3]) 5 [6]
deque([]) 3 [4]
deque([]) 4 [6]
deque([]) 6 [7]
deque([]) 7 []
1 2 5 3 4 6 7
```