

그래프 이론.

- DFS, BFS
 - 최단 경로
- => 그래프 알고리즘의 유형.

[그래프 트리 : 차이점 유의.

[인접 행렬 인접 리스트 : 메모리 사용량. 성능. 에서의 차이.

① 서로소 집합. : 공통 원소가 없는 두 집합.

↳ 서로소 관계.

[Union (합집합) : 두 집합을 하나의 집합으로 합치는 연산.

[find (탐색) : 특정 원소가 속한 집합을 알려주는 연산.

=> union - find 자료구조 라고 부른다.

ex) ① ② ③ ④ ⑤ ⑥

→ 연산 union 1,4

union 2,4

부모 노드를 리턴.

union 2,3

union 5,6

=> ① ② ③ ④ ⑤ ⑥

2개 집합.

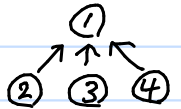
테이블

①	②	③	④	⑤	⑥
1	1	2	1	5	5

↳ 루트 번호를 재귀적으로 찾는다 => $O(V)$.

비효율적!

★ 경로를 압축 방법 =>



=> 노드간 연결성 확인

=> 그래프에서의 사이클 판별.

[무방향 그래프 → 서로소 집합 (Union-Find)

[방향 그래프 → DFS.

union 연산

→ 두 노드의 루트 노드가 같다면.

=> 사이클

② 신장 트리. (Spanning Tree)

L 그래프에 대하여, 모든 노드를 포함하면서 사이클이 존재하지 않는 부분 그래프.

→ 트리의 변형 조건.

→ 모든 간선의 가중치의 합이 '최소'가 되는 신장 트리

=> 최소 신장 트리

MST.

최소 신장 트리를 찾는 대표 알고리즘.

=> 크루스칼 알고리즘
Kruskal

- 그리디

• 크루스칼 알고리즘.

1. 간선 데이터를 가중치를 기준으로 오름차순 정렬 : $O(E \cdot \log E)$

(반복) 2. 간선을 증가씩 확인. : $O(E)$

→ (사이클 O : 무지
사이클 X : 포함)

=> $O(E \cdot \log E)$

③ 위상 정렬 (Topology Sort)

L 방향 그래프의 모든 노드를 방향성에 거스르지 않도록 정렬.

순서가 있는 정렬.

(선호관계)

ex)



Indegree (진입차수)

=> 특정 노드에 들어오는 간선의 개수.

Indegree = 2

1. Indegree가 0인 노드를 큐에 Push.

2. 큐가 빌때 까지 반복.

→ 정렬된 순서

i) 큐에서 Pop → 해당 노드에서 출발하는 간선을 그래프에서 제거.

ii) Indegree가 0이 된 노드를 큐에 Push.

* 모든 원소가 방문하기 전에 종료 => 사이클 존재.

모든 노드 방문 $O(V)$

모든 간선 처리 $O(E)$

=> $O(V+E)$