09

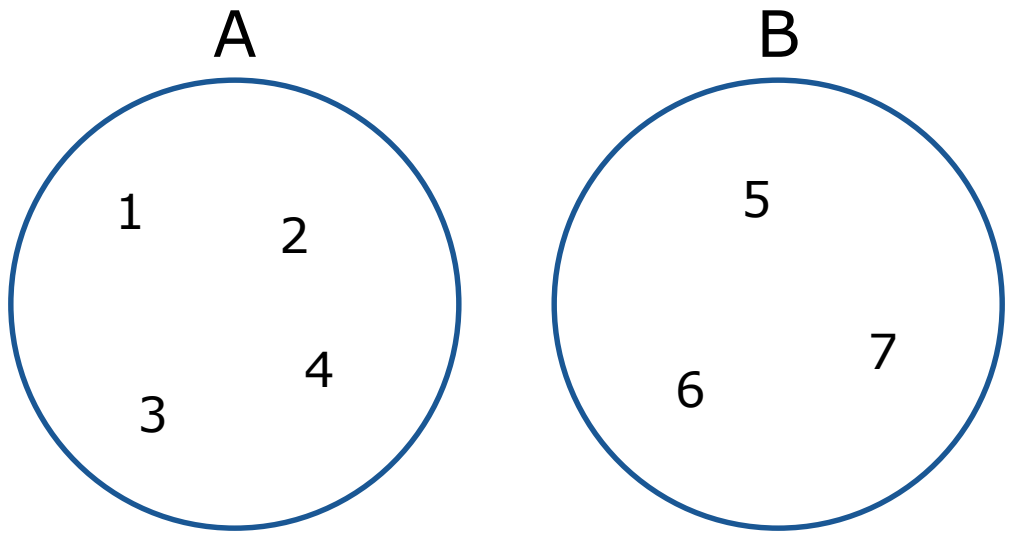
추가 알고리즘

01 유니온 파인드

02 최소 신장 트리

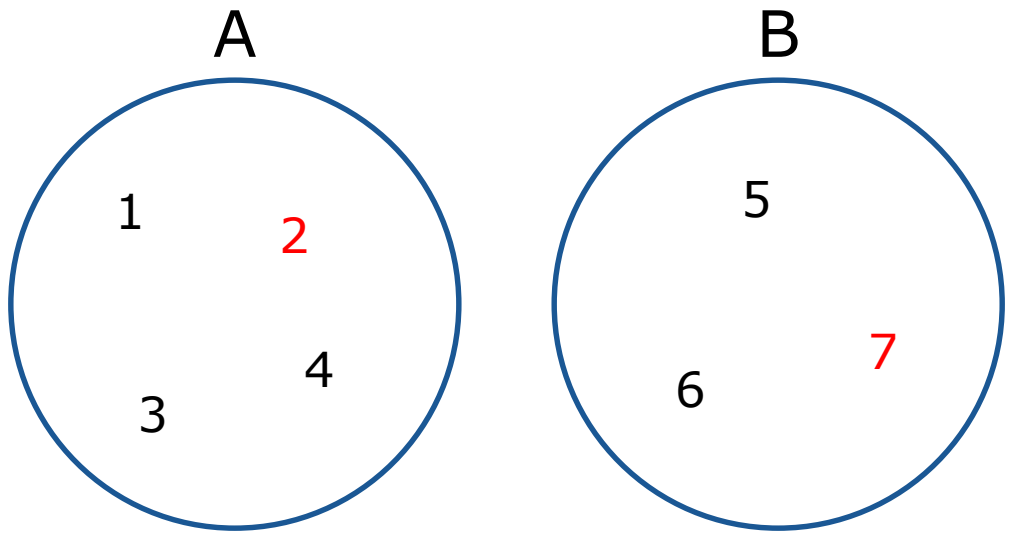
유니온 파인드

- 유니온 파인드(union-find) 알고리즘
  - union 연산
    - 특정 두 노드가 속한 집합을 묶는 연산
    - union(3, 6) → A ∪ B
  - find 연산
    - 두 노드가 같은 집합에 속해 있는지 확인하는 연산
    - find(1, 3) → true
    - find(1, 7) → false



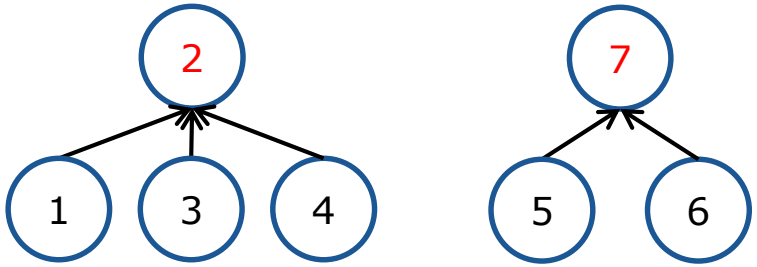
유니온 파인드

- 서로소 집합(Disjoint sets)
  - 공통 원소가 없는 두 집합 (A ∩ B = Ø)
- 대표 노드(find 연산)
  - 각 집합은 유일한 대표 노드를 가짐
    - find(1) → 2
    - find(2) → 2
    - find(3) → 2
    - find(4) → 2
    - find(5) → 7
    - find(6) → 7
    - find(7) → 7

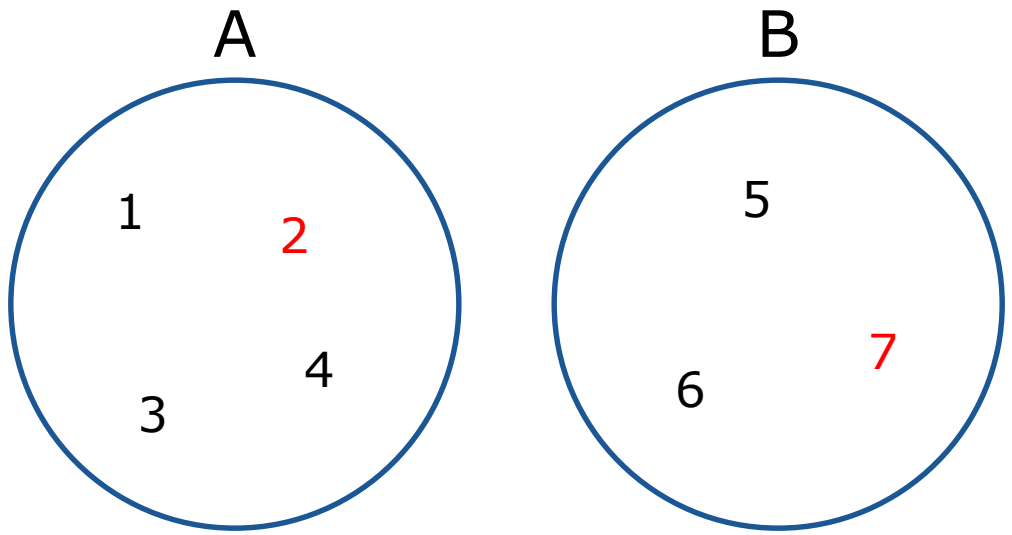


유니온 파인드

- 각 집합은 부모 노드를 갖는 트리 구조
  - 루트 노드가 대표 노드
- 부모 노드를 가리키는 1차원 배열로 표현

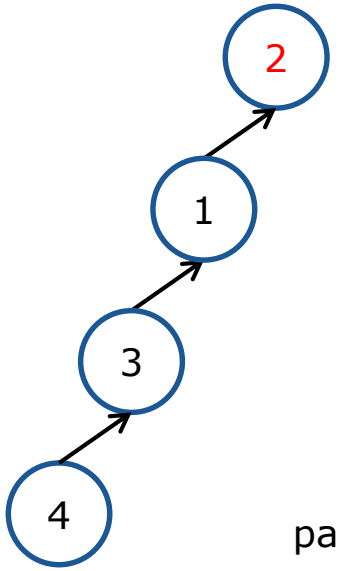


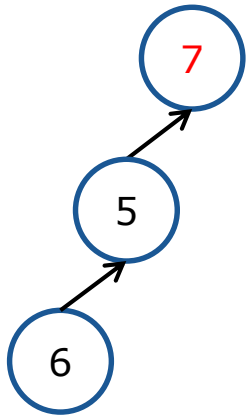
	1	2	3	4	5	6	7
parents		2	2	2	2	7	7	7



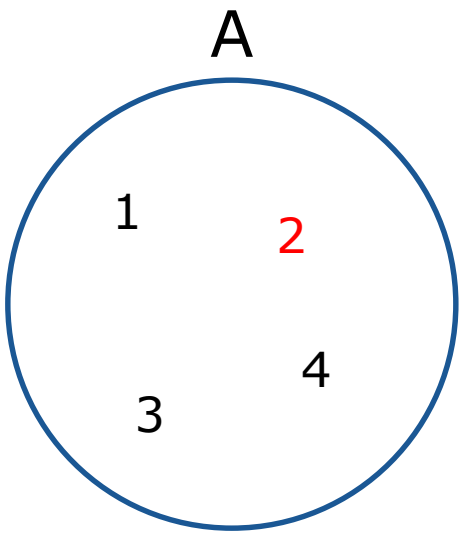
유니온 파인드

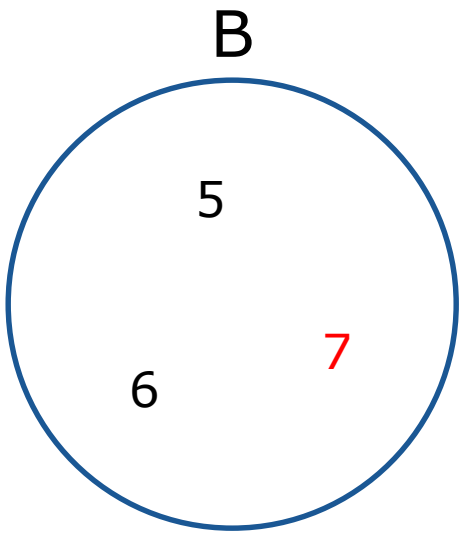
- 각 집합은 부모 노드를 갖는 트리 구조
  - 루트 노드가 대표 노드
- 부모 노드를 가리키는 1차원 배열로 표현





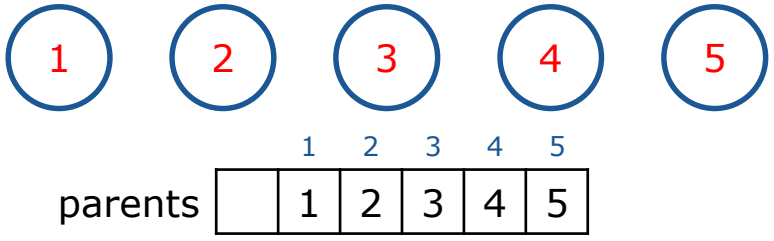
	1	2	3	4	5	6	7
parents	2	2	1	3	7	5	7





유니온 파인드

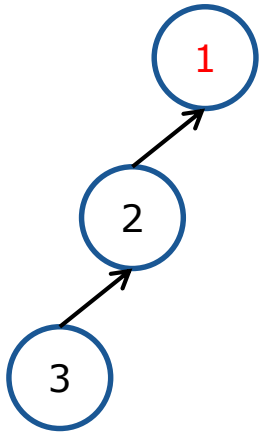
- 유니온 파인드 알고리즘 구현
  - Step1. 노드의 수 만큼 서로소 집합 만들기
    - {1}, {2}, {3}, {4}, {5}
    - 모든 노드가 대표 노드

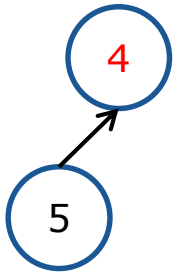


	1	2	3	4	5
parents	1	2	3	4	5

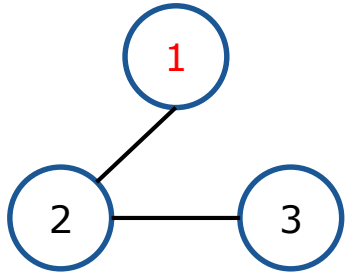
유니온 파인드

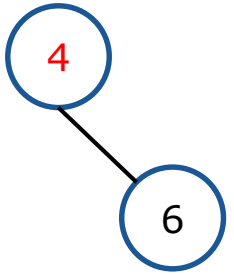
- 유니온 파인드 알고리즘 구현
  - Step2. union(1, 2) → union(2, 3) → union(4, 5) 수행
    - {1, 2, 3}, {4, 5}
    - union(x, y) : parents[y의 대표 노드] = x





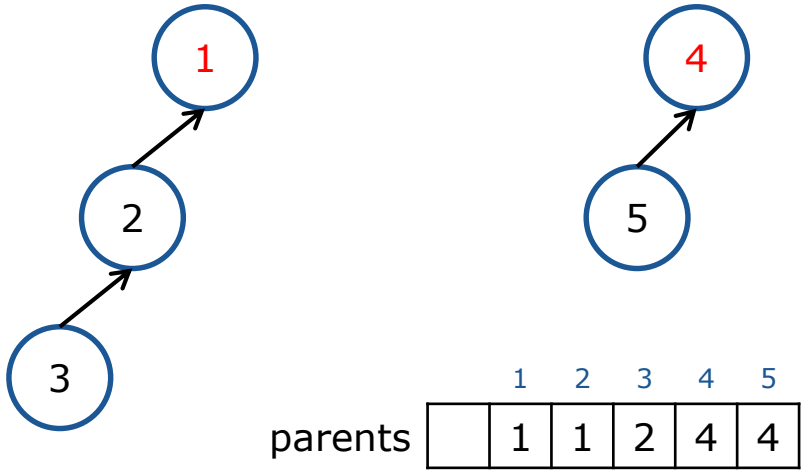
	1	2	3	4	5
parents	1	1	2	4	4





유니온 파인드

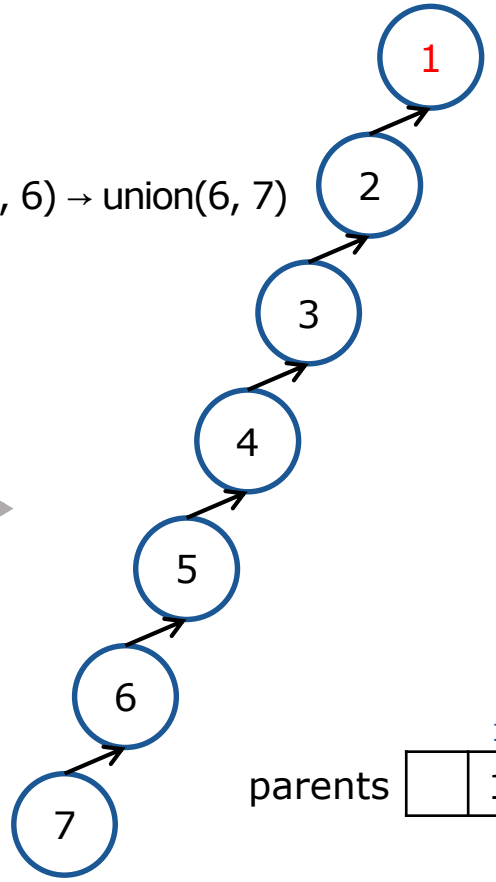
- 유니온 파인드 알고리즘 구현
  - Step3. find 연산 구현
    - {1, 2, 3}, {4, 5}
    - find(x) : x의 대표 노드(루트 노드)



유니온 파인드

- 유니온 파인드 알고리즘 최적화 – skewed tree 방지
  - 다음과 같은 연산 수행 시 skewed tree 생성
    - union(1, 2) → union(2, 3) → union(3, 4) → union(4, 5) → union(5, 6) → union(6, 7)
    - union(x, y) : parents[y의 대표 노드] = x
    - find 연산의 시간 복잡도 O(N)

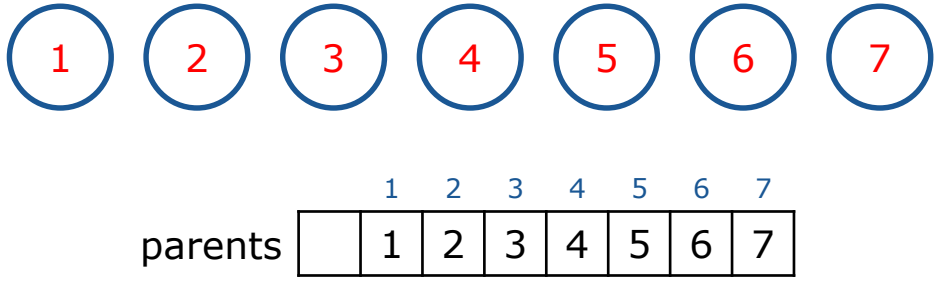
	1	2	3	4	5	6	7
parents		1	2	3	4	5	6	7

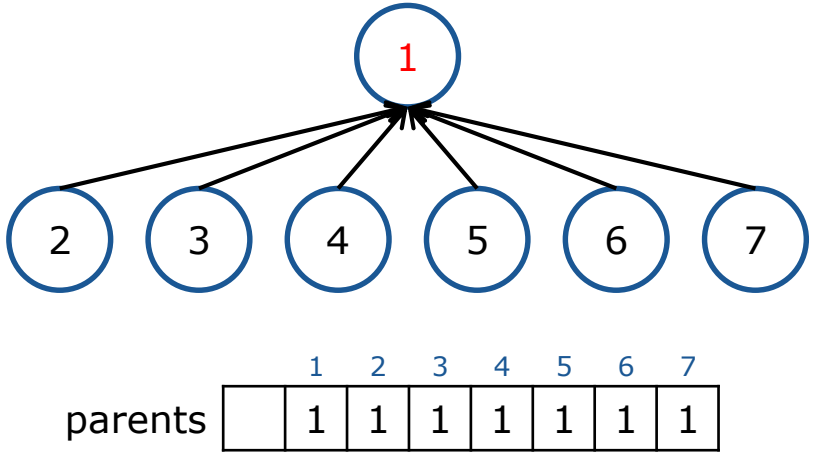


	1	2	3	4	5	6	7
parents		1	1	2	3	4	5	6

유니온 파인드

- 유니온 파인드 알고리즘 최적화 – skewed tree 방지
  - union(x, y) : parents[y의 대표 노드] = parents[x의 대표 노드] 로 연산
    - union(1, 2) → union(2, 3) → union(3, 4) → union(4, 5) → union(5, 6) → union(6, 7)
    - find 연산의 시간 복잡도 O(1)

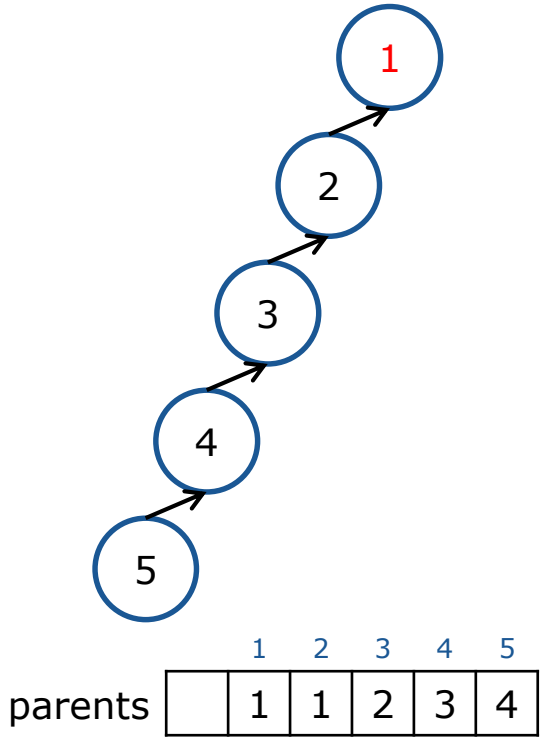


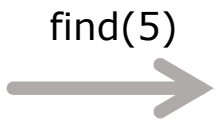


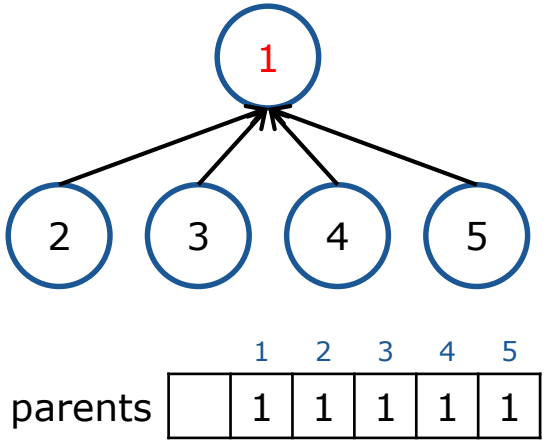
유니온 파인드

- 유니온 파인드 알고리즘 최적화 – skewed tree 방지
  - find(x) 재귀 호출 시 모든 노드의 부모 노드를 대표 노드로 변경

```
find(x) {
    if parents[x] == x then return x
    return parents[x] = find(parents[x])
}
```







연습문제

- 0부터 6까지 7개의 노드를 크기가 1인 서로소 집합을 생성하고 다음 연산을 수행하세요.
  - union(0, 1)
  - union(1, 2)
  - 1과 2가 같은 집합인지 판단
  - union(3, 4)
  - union(5, 6)
  - 4와 6이 같은 집합인지 판단

연습문제

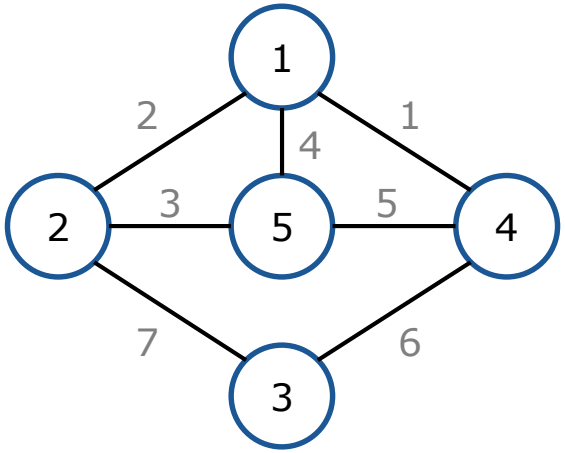
- 백준 온라인 저지 – 1717 집합의 표현

연습문제

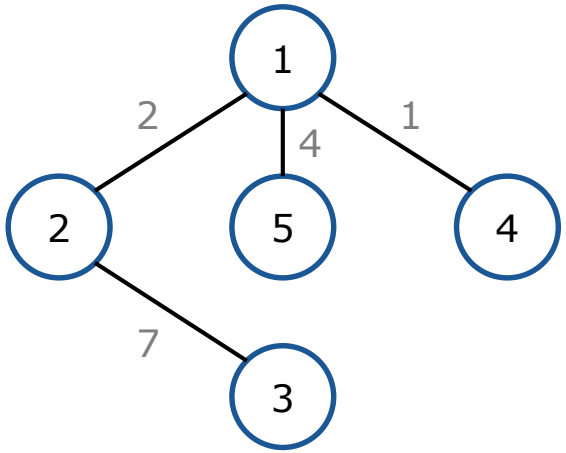
- 백준 온라인 저지 – 1976 여행 가자

최소 신장 트리

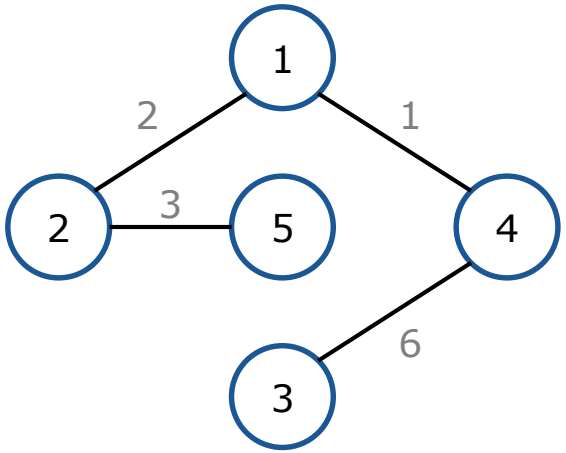
- 신장 트리(Spanning Tree)
  - 그래프의 모든 정점을 포함하는 트리
  - 간선의 개수 : V-1 개



- 최소 신장 트리(Minimum Spanning Tree, MST)
  - 신장 트리 중 가중치의 합이 최소인 트리



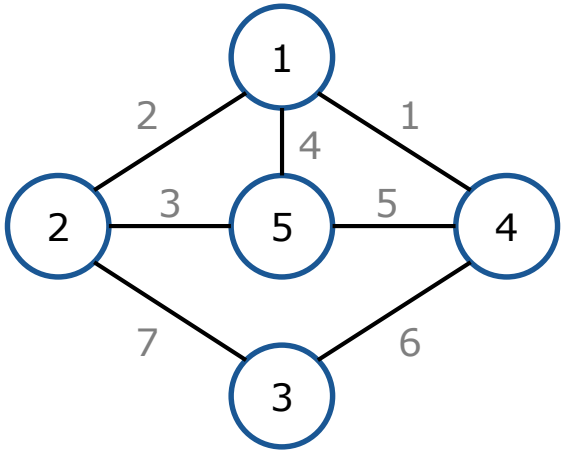
Spanning Tree



Minimum Spanning Tree(MST)

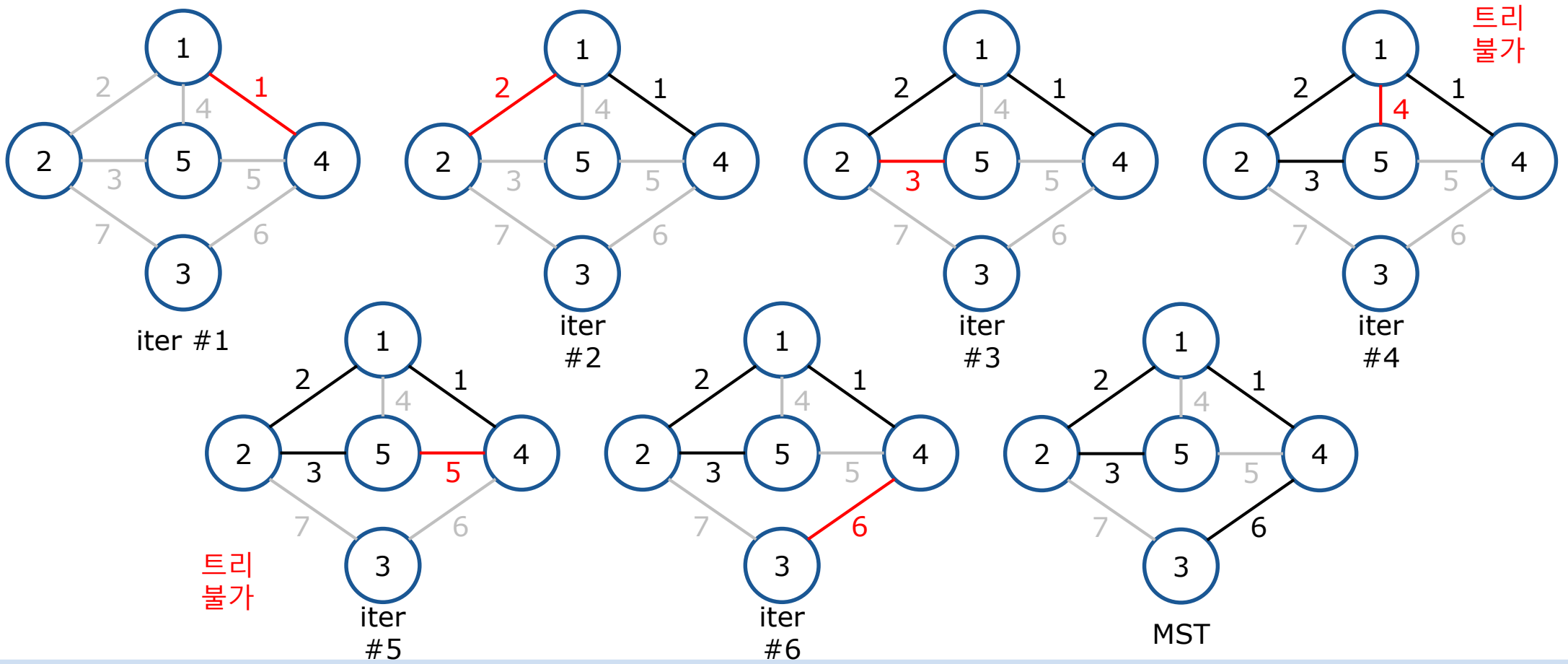
크루스칼 알고리즘

- 크루스칼(Kruskal) 알고리즘
  - 최소 신장 트리(MST)를 찾는 탐욕(greedy) 알고리즘
  - 트리 조건을 유지하며 가중치가 작은 간선부터 선택
  - 신장 트리의 간선의 개수 : V-1 개



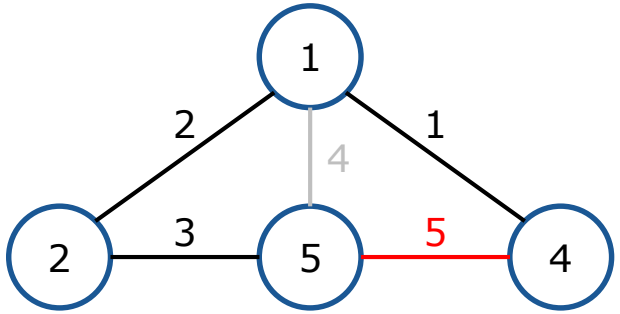
크루스칼 알고리즘

- 크루스칼(Kruskal) 알고리즘



크루스칼 알고리즘

- 크루스칼(Kruskal) 알고리즘
  - 트리 조건 : 사이클(Cycle) 발생 x
  - 유니온 파인드 알고리즘을 사용하여 사이클 발생 여부 판단
    - 추가할 간선의 두 정점이 같은 집합이면 사이클 발생



연습문제

- 백준 온라인 저지 – 1197 최소 스패닝 트리