

# 최소 신장 트리

- 01 크루스칼 메소드
- 02 프림 메소드
- 03 최소 신장 트리 문제 풀이

신 제 용

# 01 크루스칼 메소드

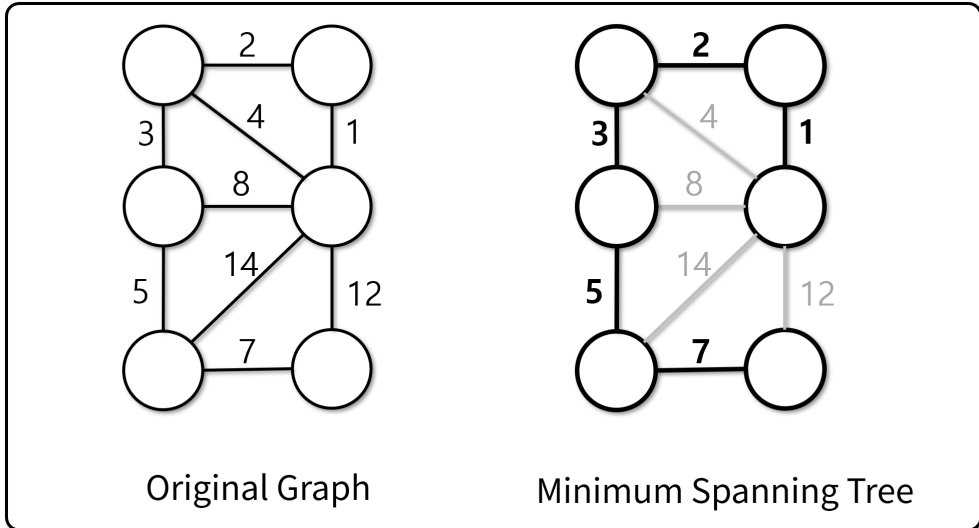
최소 신장 트리를 구하는 방법 중 하나인 크루스칼 방법을 배웁니다.

학습 키워드 – 크루스칼, Kruskal, 유니온-파인드, Union-Find, MST

Chapter 01  
크루스칼 메소드

# 최소 신장 트리

- Minimum Spanning Tree (MST)
- 그래프의 모든 노드를 최소 가중치 합으로 연결한 부분 트리
  - 부분 트리어므로, **회로(Cycle)**가 존재해서는 안된다. (Acyclic)



Chapter 01

크루스칼 메소드

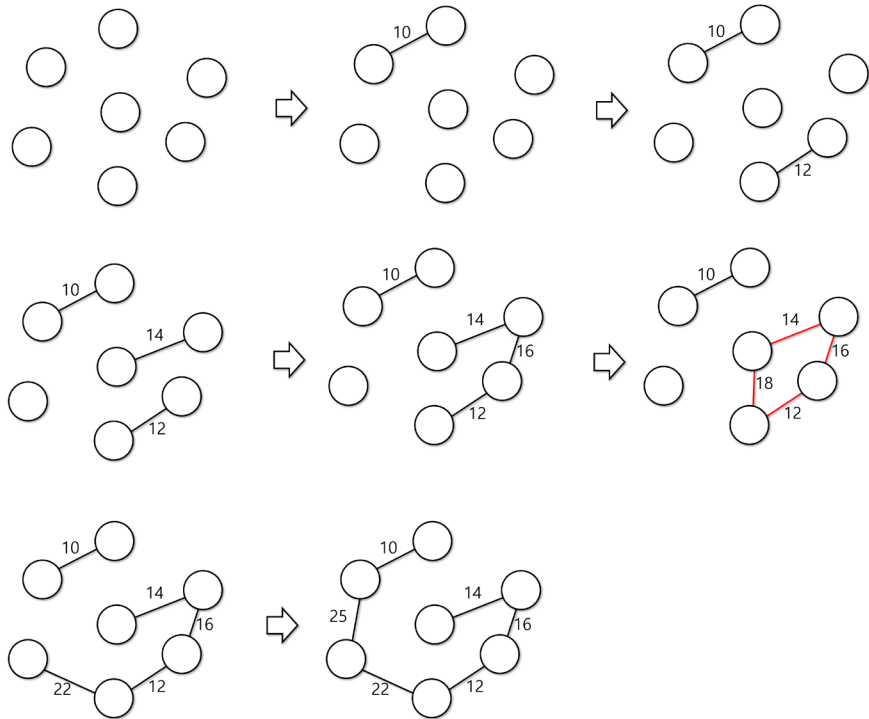
# 크루스칼 메소드 (Kruskal's Method)

- 모든 간선을 가중치를 기준으로 오름차순 정렬 ( $O(E \log E)$ )
  - 모든 간선을 정렬하는 과정때문에, **간선의 수가 적을 때 사용**
- 작은 가중치의 간선부터 하나씩 선택하여 MST를 구성한다.
  - 이 과정에서 사이클을 형성하는 간선은 선택하지 않는다.
  - 사이클 형성을 판단하는 알고리즘은 **Union-Find 알고리즘** 사용

Chapter 01

크루스칼 메소드

# 크루스칼 메소드 예시



Kruskal's Algorithm

## Chapter 01 크루스칼 메소드

# 유니온-파인드 (Union-Find)

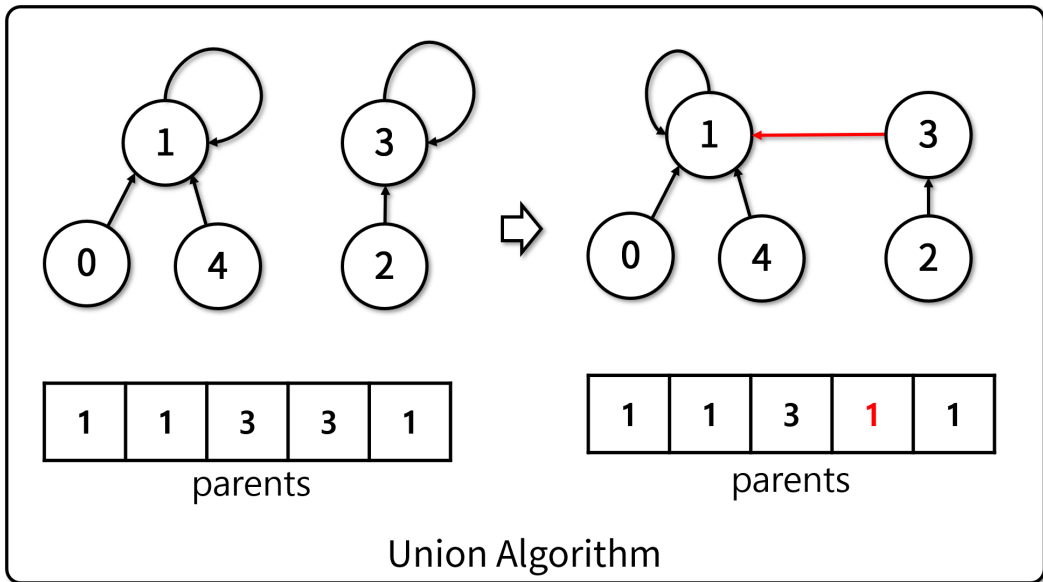
- 트리 형식으로 집합을 만들고 연산하는 알고리즘
  - 유니온 (Union) – 두 집합의 합집합을 계산하는 연산
  - 파인드 (Find) – 한 원소가 속한 집합을 알아내는 연산  
(트리에서 루트 노드를 찾는 연산)
- 크루스칼 방법과 유니온-파인드 알고리즘
  - 초기에 개별 노드가 크기가 1인 집합을 이룬다.
  - MST에 **노드를 하나 추가할 때 마다 해당 노드를 유니온 연산한다.**
  - 노드를 추가할 때 **파인드 연산으로 같은 집합의 원소인지 확인**  
(사이클이 발생하는지 확인)한다.

Chapter 01

크루스칼 메소드

# 유니온 알고리즘

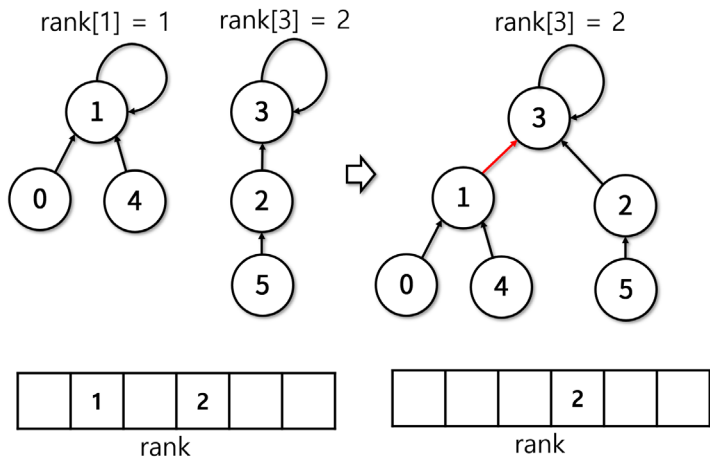
- 각 노드는 부모 노드를 parents 배열에 기록한다.
  - 루트 노드는 자기 자신을 부모 노드로 가진다.
- Union: 두 집합 중 하나의 루트 노드를 다른 루트 노드의 부모로 한다.



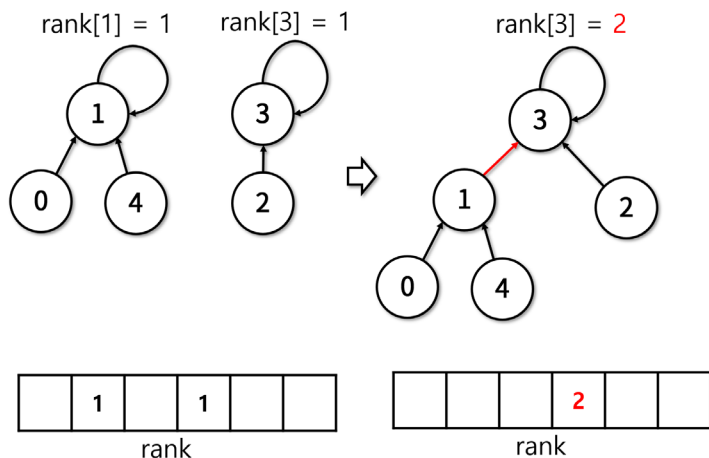
Chapter 01  
크루스칼 메소드

# Union-By-Rank

- 유니온 이후 집합의 랭크(트리의 깊이)를 최소한으로 유지하는 방법
  - 랭크가 더 높은 집합이 루트가 되게 한다.
  - 랭크가 같을 경우 아무거나 선택하고, 랭크가 1 증가한다.



Union-By-Rank



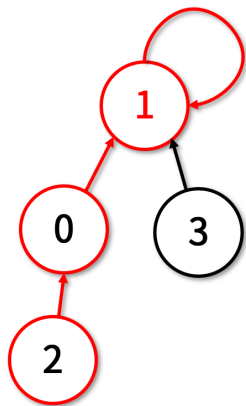
두 집합의 rank가 같을 때의 Union-By-Rank

Chapter 01  
크루스칼 메소드



# 파인드 알고리즘

- 해당 집합의 루트 노드를 반환. 재귀적으로 동작하여 자기 자신이 부모일 때 까지 동작



$\text{find}(x) = \text{find}(\text{parents}[x])$

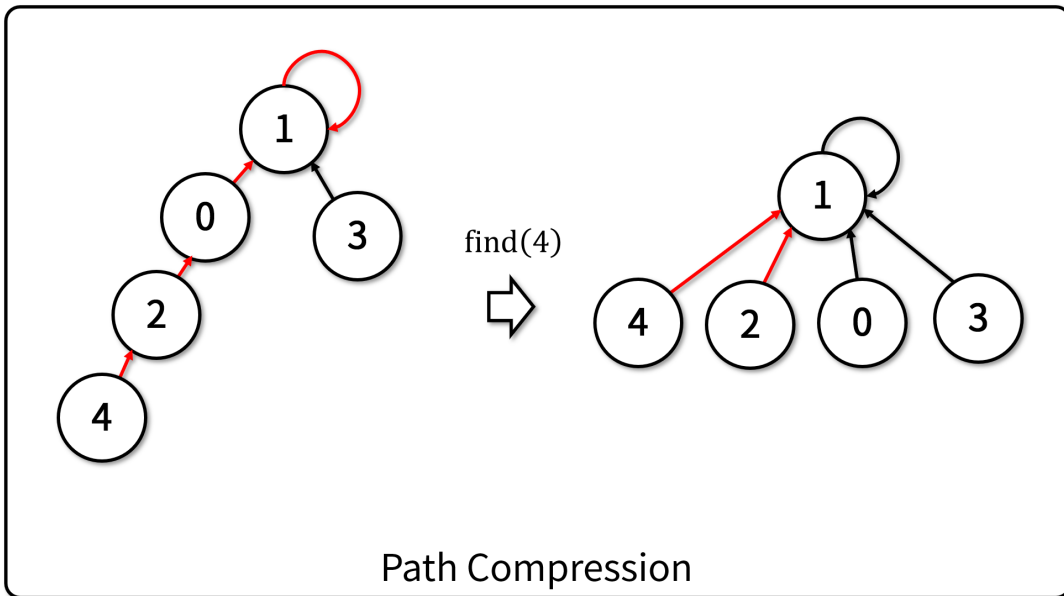
$\text{find}(2) = \text{find}(0) = \text{find}(1) = 1$

Find Algorithm

Chapter 01  
크루스칼 메소드

# 경로 단축 (Path Compression)

- Find 과정에 참여한 모든 노드의 부모를 루트 노드로 변경
  - 한번 Find를 동작시키면 여러 노드의 Find가  $O(1)$ 로 동작



Chapter 01  
크루스칼 메소드

## 02 프림 메소드

다음 챕터에서는 최소 신장 트리를 생성하는 또 다른 방법인  
프림 메소드를 학습합니다.

Chapter 01

크루스칼 메소드

# 02 프림 메소드

MST를 구성하는 또 다른 알고리즘인 프림 메소드를 학습합니다.

학습 키워드 – 프림 메소드, Prim's method, MST

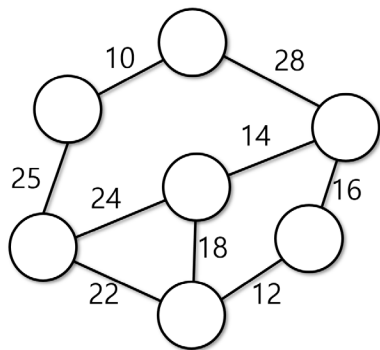
Chapter 02  
프림 메소드

# 프림 메소드 (Prim's method)

- 임의의 노드에서 시작하여 하나씩 노드를 연결하는 방식
- 연결된 모든 노드의 간선 중 가장 낮은 가중치의 간선을 선택 (힙으로 최적화)
- 간선의 개수가 많을 경우 크루스칼 메소드보다 유리
- 시간복잡도:  $O(E \log V)$

Chapter 02  
프림 메소드

# 프림 메소드 예시



heap							
connected							
mst							
weights							

Chapter 02  
프림 메소드

# 03 최소 신장 트리 문제 풀이

다음 챕터에서는 최소 신장 트리를 구현하는 방법을 학습합니다.

Chapter 02  
프림 메소드

# 03 최소 신장 트리 문제 풀이

최소 신장 트리(MST)를 구현하고, 이를 활용하는 문제를 해결해 봅시다.

학습 키워드 – 최소 신장 트리, MST, 유니온-파인드, 구현

## Chapter 03

### 최소 신장 트리 문제 풀이



# Problem1

---

## 문제 설명

---

Kruskal's Method를 구현하세요!

전체 노드의 개수 `N` 개가 있고, 각 노드간의 연결이 `edge[i] = [출발노드, 도착노드, 가중치]` 로 주어져 있다고 하자.

이 때 최소 신장 트리를 구성하기 위한 가중치의 총 합을 구하시오.

## 매개변수 예시

---

`N = 5`

`edge = [[1, 2, 2], [1, 3, 3], [2, 3, 4], [2, 4, 5], [3, 4, 6], [5, 1, 1]]`

## 출력 예시

---

11

## Chapter 03

### 최소 신장 트리 문제 풀이

## Problem2

---

### 문제 설명

---

Prim's Method를 구현하세요!

전체 노드의 개수 `N` 개가 있고, 각 노드간의 연결이 `edge[i] = [출발노드, 도착노드, 가중치]` 로 주어져 있다고 하자.

이 때 최소 신장 트리를 구성하기 위한 가중치의 총 합을 구하시오.

### 매개변수 예시

---

```
N = 5
```

```
edge = [[1, 2, 2], [1, 3, 3], [2, 3, 4], [2, 4, 5], [3, 4, 6], [5, 1, 1]]
```

### 출력 예시

---

```
11
```

## Chapter 03

### 최소 신장 트리 문제 풀이

## Problem3

### 문제 설명

당신에게 2차원 평면상의 좌표  $x[i]$  와  $y[i]$  가 주어진다.

각 좌표에 찍혀있는 점을 서로 연결하는 데에는 두 좌표 사이의 '맨하탄 거리' 만큼의 비용이 든다.

$i$  번째 점과  $j$  번째 점 사이의 맨하탄 거리는 아래와 같이 정의된다.

$$\text{manhattan}(i, j) = |x[i] - x[j]| + |y[i] - y[j]|$$

이 때, 모든 점을 연결하는 데에 필요한 최소의 비용을 구하시오.

### 매개변수 형식

$x = [0, 0, 3, 3, 6]$

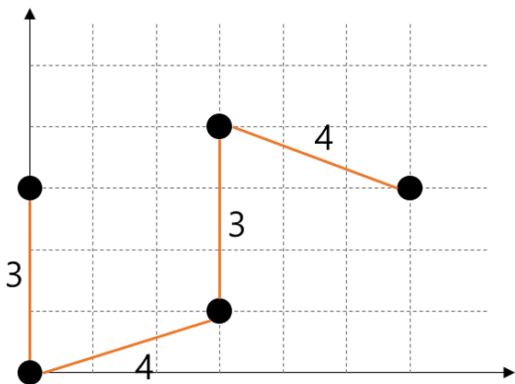
$y = [0, 3, 1, 4, 3]$

### 반환값 형식

14

#### 입출력 설명

아래와 같이 연결했을 때 최소의 비용이 된다.



## Chapter 03

### 최소 신장 트리 문제 풀이

# Problem4

## 문제 설명

당신은 좀비 바이러스 치료제를 단 하나 가지고 있다.

이번에 매우 전염성이 높은 좀비 바이러스가 퍼져, 긴급히 방역이 필요한 상황이다.

총  $N$  명의 인원이 관리 대상으로,  $i$  번째 인원과  $j$  번째 인원이 서로 가까이 있어 감염시킬 수 있는 경우  $graph[i][j]$  가 1로 주어진다.

서로 가까이 있는 인원 중에 한 명이라도 감염된 인원이 있다면, 결국 모두 서로를 감염시키게 된다.

현재 좀비 바이러스에 감염된 인원은 `infected` 배열에 주어진다.

당신은 치료제가 단 하나 있기 때문에, `infected`의 인원 중 한 명을 치료할 수 있다.

이 때, 어떤 인원을 치료해야 좀비 바이러스에 감염되는 인원을 최소화할 수 있는지 해당 인원의 인덱스를 출력하시오.

단, 정답이 여럿인 경우 더 작은 인덱스를 출력하시오.

## 매개변수 형식

`N = 3`

```
graph = [[1, 1, 0],
          [1, 1, 0],
          [0, 0, 1]]
```

`infected = [0, 2]`

## 반환값 형식

`0`

## 예시입출력 설명

`0` 번 인원을 치료할 경우 2명을 감염을 막을 수 있으나, `2` 번 인원을 치료할 경우 1명만 감염을 막을 수 있다.

## Chapter 03

### 최소 신장 트리 문제 풀이

