

## Day 6 - 그리드(Greed) 알고리즘

DFS / BFS ⇒ 완전 탐색 (1문제)

그리드(1문제)

1. 크루스칼 알고리즘
2. 프림 알고리즘
  - 탐욕 알고리즘
    - a) 어떠한 결정 순간마다 눈 앞에 보이는 것을 최선의 선택이라고 생각하고 그 결정에 따른다.
    - b) 그러나 이것은 비효율적이다. (따라서 크루칼, 프림 알고리즘을 쓴다)
    - c) 결과값을 신뢰할 수 없다.

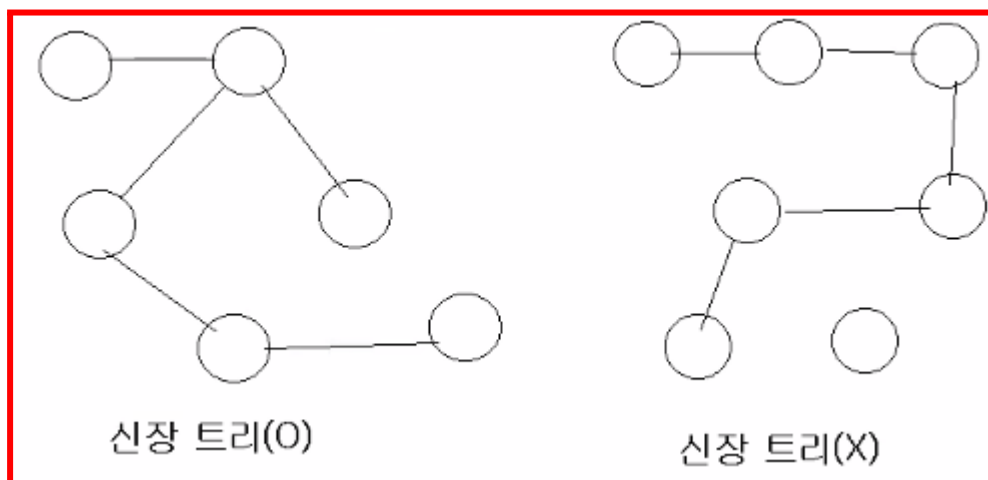
Ex) 5원, 2원, 1원 ⇒ 8원을 거슬러 주어야 한다.

- 5원 → 1개
- 2원 → 1개
- 1원 → 1개
  
- 5원 → 1개
- 1원 → 3개

그리드 알고리즘을 사용하는 대표적인 알고리즘이 존재한다

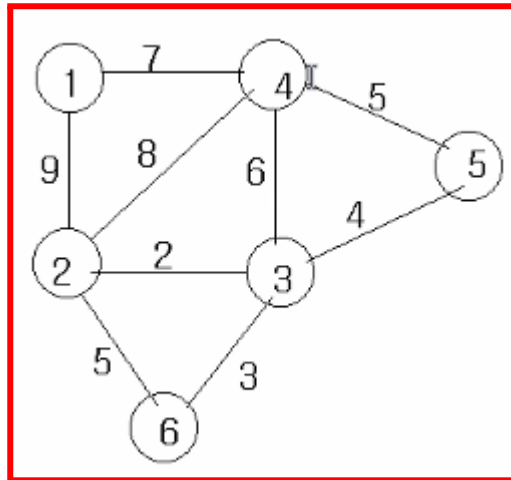
### 1) 최소 신장 트리

- a) 최소 + 신장 트리
- b) 신장 트리의 조건
  - i) 모든 정점(노드)은 간선으로 연결
  - ii) 간선의 개수는 정점의 개수보다 하나 적다.



c) 최소 신장 트리

- i) 가중치의 합이 최소
- ii) 단점: (결과값이 하나 이상 존재할 수 있다)
  - (1) 1 부터 3까지 가는 간선의 가중치의 합이 13으로 같은 값이 두개 이상이 나올 수 있다. (같은 목적지로 가는 값이 같은게 두개 이상 나올 수 있다).
- iii) 가중치가 존재한다.



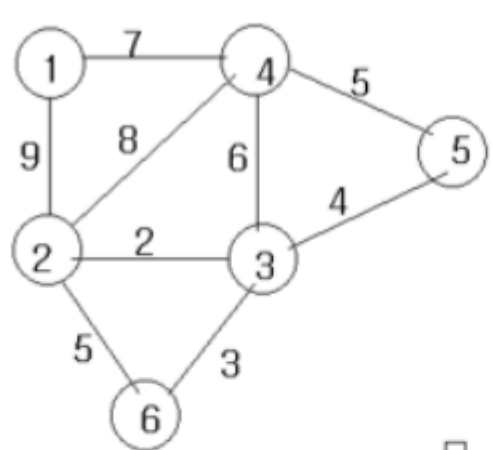
d) 최소 신장 트리를 푸는 알고리즘

- i) 크루스칼 알고리즘
- ii) 프림 알고리즘

## 크루스칼 알고리즘

- 1) 간선의 가중치를 기준으로 오름 차순 정렬
  - a) 가장 작은 값이 위로 오게 만든다
- 2) 시작은 가중치가 낮은 간선부터 시작되어 하나씩 신장 트리에 추가한다.
  - 순환이 생기지 않는다면 최소 신장 트리에 추가하고 순환이 생기면 추가하지 않는다.

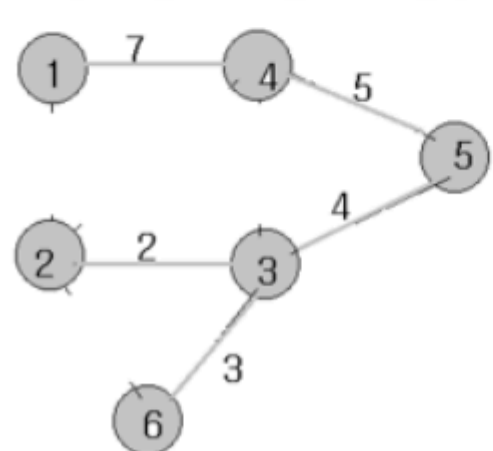
(순환이 생기지 않는다면 최소 신장 트리에 추가하고 순환이 생기면 추가하지 않는다)



| 시작 | 도착 | 가중치 |
|----|----|-----|
| 2  | 3  | 2   |
| 3  | 6  | 3   |
| 3  | 5  | 4   |
| 2  | 6  | 5   |
| 4  | 5  | 5   |
| 3  | 4  | 6   |
| 1  | 4  | 7   |
| 2  | 4  | 8   |
| 1  | 2  | 9   |

↓

1. 가중치가 가장 작은 2 - 3을 최소 신장 트리에 추가



| 시작 | 도착 | 가중치 |
|----|----|-----|
| 2  | 3  | 2   |
| 3  | 6  | 3   |
| 3  | 5  | 4   |
| 2  | 6  | 5   |
| 4  | 5  | 5   |
| 3  | 4  | 6   |
| 1  | 4  | 7   |
| 2  | 4  | 8   |
| 1  | 2  | 9   |

## 그리드 알고리즘 문제 유형

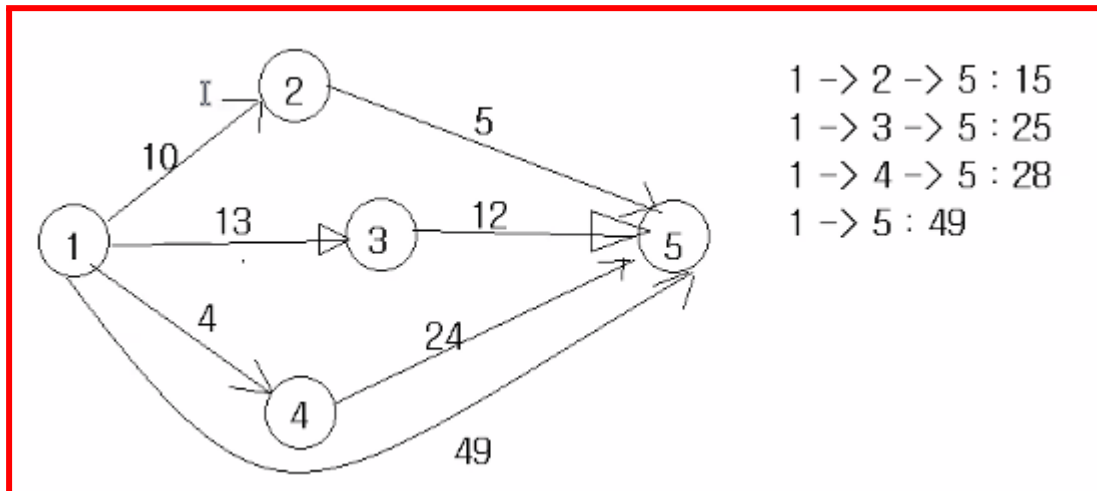
1. 배낭
2. 집합 문제

## 프림 알고리즘

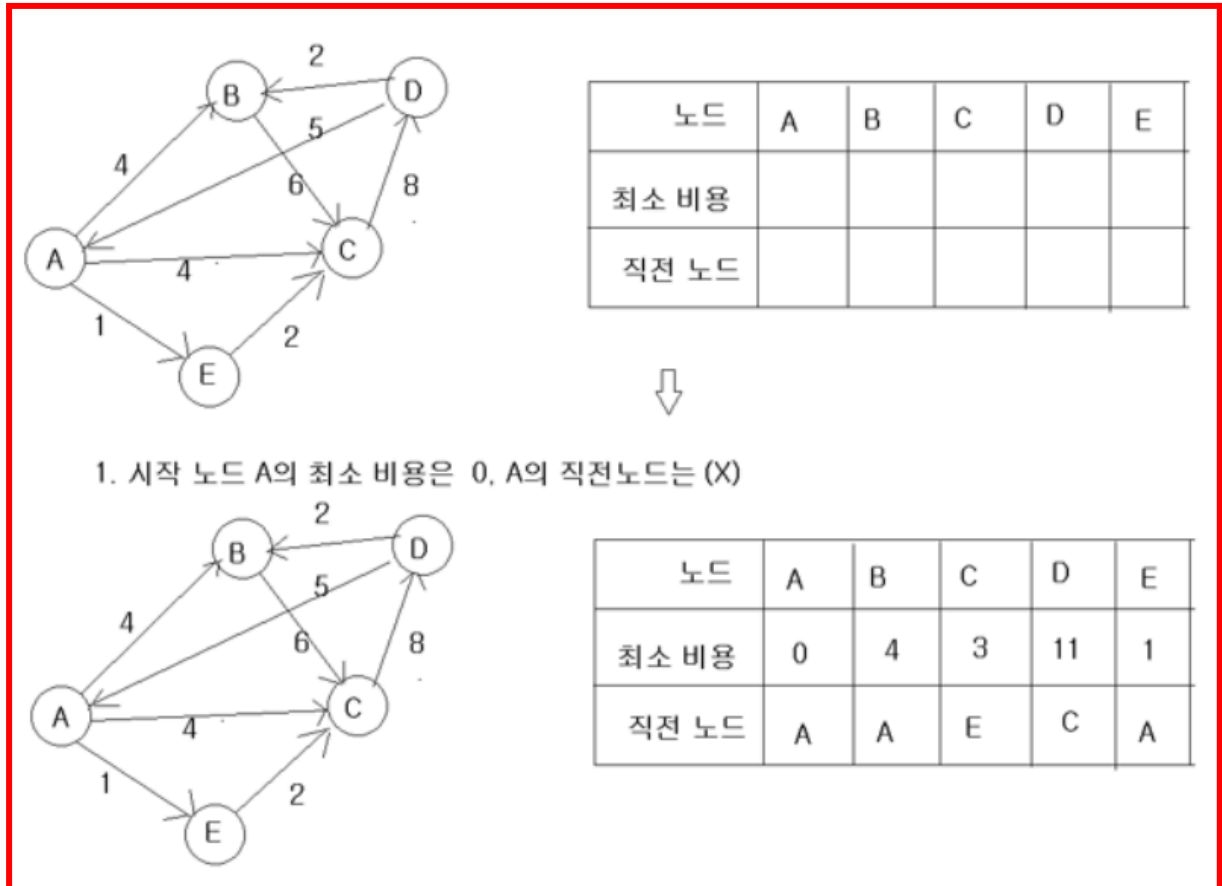
- 프림 알고리즘은 매 순간 최선의 조건을 선택하는 그리디 알고리즘을 바탕으로 한다. 즉, 탐색노드에 대해 연결된 인접 노드 중 비용이 가장 적은 간선으로 연결된 노드를 선택한다.

## 다익스트라 알고리즘

1. 최단거리(가중치의 합)
2. 그리드와 달리 순환이 가능하고 간선, 노드의 수가 상관이 없다.



3. 기본 시작점은 (1,1) 또는 문제에서 지정을 해준다.



## 벨만 포드 알고리즘

- 가중치가 음수값을 가질 수 있다.

## 무슨 알고리즘을 써야 하나?

- 가중치가 있는 그래프
  1. 그리드
    - a. 제일 처음에는 그리드로 접근한다 → 다익스트라
  2. 다익스트라
    - a. 문제유형이 많이 나온다 (최단거리 구하는 문제)
    - b. 순환문제가 나오면 다익스트라 사용