

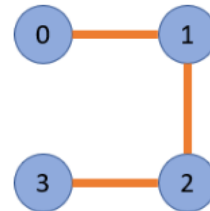
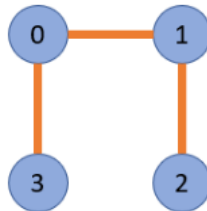
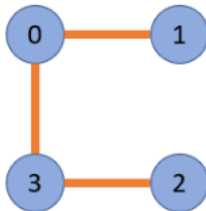
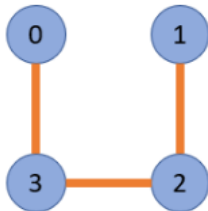
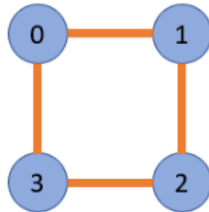
# Spanning Tree, Dijkstra

## 1. Spanning Tree

### Spanning Tree (신장 트리)

모든 정점을 포함하고, 정점간 서로 연결되어있지만 사이클이 존재하지 않는 그래프

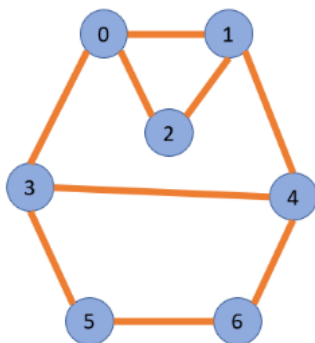
만약  $n$ 개의 정점을 가지는 그래프가 있다면  $n$ 개의 정점을 모두 연결할 수 있는  $n-1$ 개의 엣지로 이루어진 부분 그래프는 모두 신장 트리가 될 수 있다. 다음은 신장 트리의 예시를 그림으로 나타낸 것이다.



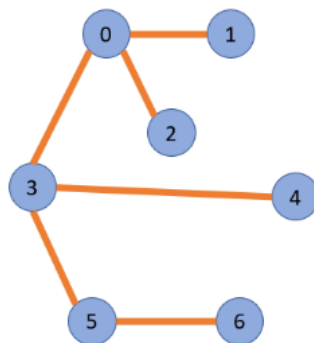
<조건>

1. 연결 그래프의 부분 그래프이며, 그래프에서 모든 정점을 포함한다
2. 모든 정점이 서로 연결이 되어 있어야 한다
3. 사이클이 존재하지 않아야 한다
4. 연결 그래프에서 신장 트리는 1개가 아닌 다수일 수 있다

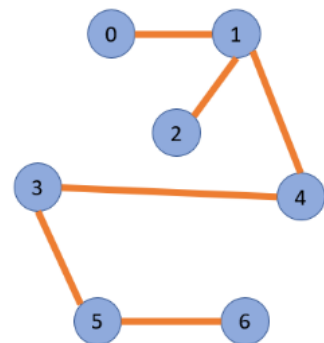
신장 트리는 깊이 우선 탐색으로 만들어진 신장 트리(depth first spanning tree)와 너비 우선 탐색으로 만들어진 신장 트리(breadth first spanning tree)로 나누어 진다. 다음은 각 트리에 대한 예시이다.



Graph G



Breadth First spanning tree



depth First spanning tree

이러한 신장 트리는 최소 신장 트리를 구현하는 베이스가 된다

## 2. Dijkstra

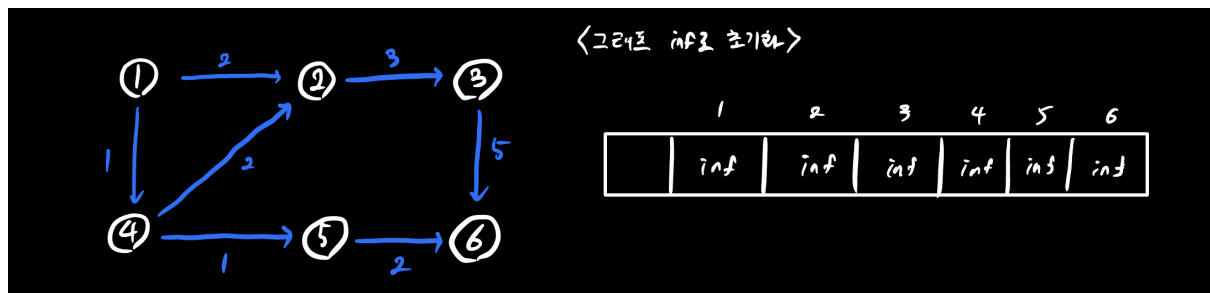
- DP이나 그리디 알고리즘을 활용한 대표적인 최단 경로 탐색 알고리즘
  - 매 상황에서 가장 비용이 적은 노드를 선택(그리디 알고리즘)
  - 특정한 하나의 정점에서 다른 모든 정점으로 가는 최단 경로를 알려준다  
플로이드 - 워셜 은  $N:N$  의 최단 경로
  - 이때, 음의 간선을 포함할 수 없다  
현실 세계에서는 음의 간선이 존재하지 않기 때문에, 인공위성 GPS 소프트웨어 등에서 많이 사용된다

알고리즘의 동작 과정

- ① 출발 노드와 도착 노드를 설정한다.
- ② '최단 거리 테이블'을 초기화한다.
- ③ 현재 위치한 노드의 인접 노드 중 방문하지 않은 노드를 구별하고, 방문하지 않은 노드 중 거리가 가장 짧은 노드를 선택한다. 그 노드를 방문 처리한다.
- ④ 해당 노드를 거쳐 다른 노드로 넘어가는 간선 비용(가중치)을 계산해 '최단 거리 테이블'을 업데이트한다.
- ⑤ ③~④의 과정을 반복한다.

'최단 거리 테이블'은 1차원 배열로,  $N$ 개 노드까지 오는 데 필요한 최단 거리를 기록한다.  $N$ 개(1부터 시작하는 노드 번호와 일치시키려면  $N + 1$ 개) 크기의 배열을 선언하고 큰 값을 넣어 초기화시킨다.

'노드 방문 여부 체크 배열'은 방문한 노드인지 아닌지 기록하기 위한 배열로, 크기는 '최단 거리 테이블'과 같다. 기본적으로는 `False`로 초기화하여 방문하지 않았음을 명시한다.



① 1번 노드에서 출발 → 거리 0으로 설정

② 인접한 2, 4번 노드 확인

|    | 1 | 2   | 3   | 4   | 5   | 6   |
|----|---|-----|-----|-----|-----|-----|
| 기준 | 0 | inf | inf | inf | inf | inf |
| 갱신 | 0 | 2   | inf | 1   | inf | inf |

③ 기준 값과 비교해서 더 작은 값으로 갱신

④ 갱신한 테이블 중 방문하지 않은 노드 중 거리가 가장 작은 번호를 다음 노드로 설정  
⇒ 4번 노드

④ 인접 노드 모두 확인했기 때문에 방문 처리

|    | 1 | 2 | 3   | 4 | 5   | 6   |
|----|---|---|-----|---|-----|-----|
| 기준 | 0 | 2 | inf | 1 | inf | inf |
| 갱신 | 0 | 2 | inf | 1 | 2   | inf |

② 갱신한 뒤 값이 같다면 숫자가 더 작은 노드를 선택

① 기준 값이 더 작으므로 갱신 되지 않음

### <다익스트라 알고리즘의 특징>

단계를 거치며 한 번 처리된 노드의 최단거리는 고정되어 더 이상 바뀌지 않는다

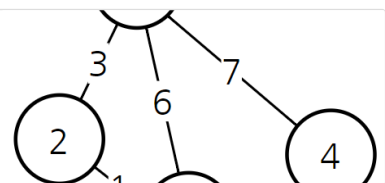
→ 한 단계 당 하나의 노드에 대한 최단 거리를 확실히 찾는 것으로 이해할 수 있다

### ▼ 관련 강의 및 참고자료

#### 23. 다익스트라(Dijkstra) 알고리즘

&nbsp; 다익스트라(Dijkstra) 알고리즘은 다이나믹 프로그래밍을 활용한 대표적인 최단 경로(Shortest P...

<https://m.blog.naver.com/ndb796/221234424646>



#### 25강 - 다익스트라 알고리즘(Dijkstra Algorithm) [ 실전 알고리즘 강좌(Algorithm Programming Tutorial) #25 ]

25강 - 다익스트라 알고리즘(Dijkstra Algorithm) [ 실전 알고리즘 강좌(Algorithm Programming Tutorial) #25 ] 강의 동영상입니다. 이번 시간에는 다익스트라 최단 경로 찾기 알고리즘에 대해 공부하는 시간을 가집니다.

<https://www.youtube.com/watch?v=611B-9zk2o4>

