9주차 2차시 최소신장 나무

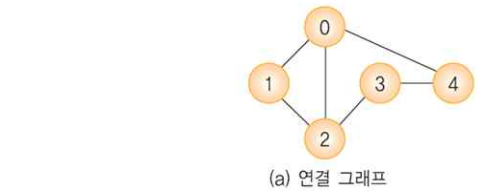
【학습목표】

1. 크로스칼의 알고리즘을 이해할 수 있다.
2. 프림 알고리즘을 이해할 수 있다.

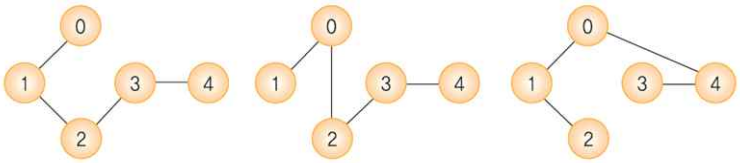
학습내용1 : 크로스칼의 알고리즘

1. 신장트리

◎ 그래프내의 모든 정점을 포함하는 트리
◎ 모든 정점들이 연결되어 있어야 하고 사이클을 포함해서는 안됨
◎ n개의 정점을 가지는 그래프의 신장트리는 n-1개의 간선을 가짐
◎ 최소의 링크를 사용하는 네트워크 구축 시 사용: 통신망, 도로망, 유통망 등



(a) 연결 그래프

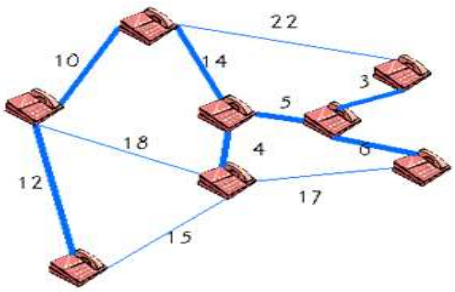


(b) 신장 트리 중의 일부

신장 트리는 깊이 우선 탐색이나 너비 우선 탐색 도중에 사용된 간선만 모으면 만들 수 있다. 위의 그림 맨 왼쪽에 있는 신장트리는 시작 정점 0으로 깊이우선 탐색을 할 때 사용된 간선으로만 만들어진 신장트리(깊이 우선 신장 트리)이고 중간 신장트리는 시작 정점을 0으로 너비 우선 탐색할 때 사용된 간선으로만 만들어진 신장트리(너비 우선 신장 트리)이다.

2. 최소 신장 나무(minimum spanning tree : MST)

◎ 네트워크에 있는 모든 정점들을 가장 적은 수의 간선과 비용으로 연결
◎ MST의 응용
- ☞ 도로 건설 - 도시들을 모두 연결하면서 도로의 길이를 최소가 되도록 하는 문제
- ☞ 전기 회로 - 단자들을 모두 연결하면서 전선의 길이를 가장 최소로 하는 문제
- ☞ 통신 - 전화선의 길이가 최소가 되도록 전화 케이블 망을 구성하는 문제
- ☞ 배관 - 파이프를 모두 연결하면서 파이프의 총 길이를 최소로 하는 문제



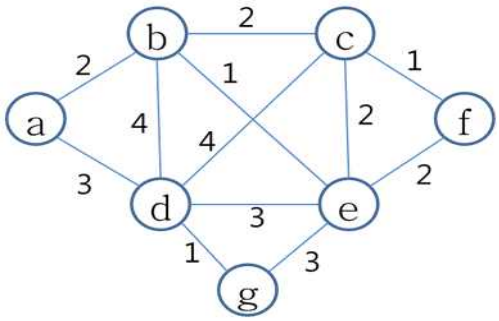
* 최소 신장 나무(minimum spanning tree)
• 그래프 G의 모든 정점들을 사이클이 생기지 않게 연결하면서 간선에 부여된 가중치의 합이 가장 적게 구성된 트리 (욕심쟁이 방법적용)
• 최소 신장 트리를 만드는 방법 : Kruskal과 Prim이 제안한 알고리즘

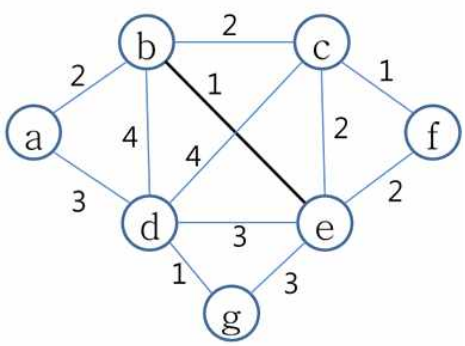
* 정의

① 신장나무: 가중 무방향 그래프에 대하여 신장나무는 주어진 그래프의 모든 노드들을 포함하는 연결된 부분 그래프로서 나무를 형성한다.

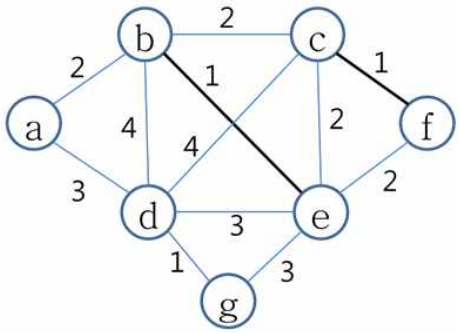
② 최소비용 신장나무: 신장나무 중에서 가중치의 합이 가장 작은 신장나무. 최소신장 나무라고도 함.

③ 최소 신장 나무 구축 과정

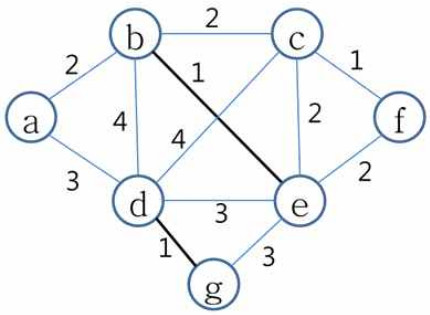




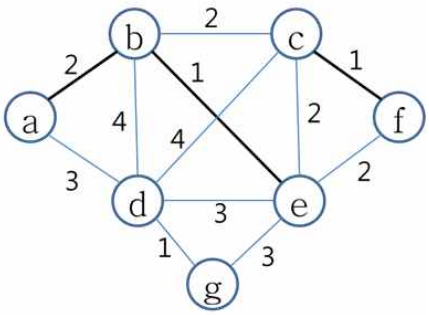
(a)



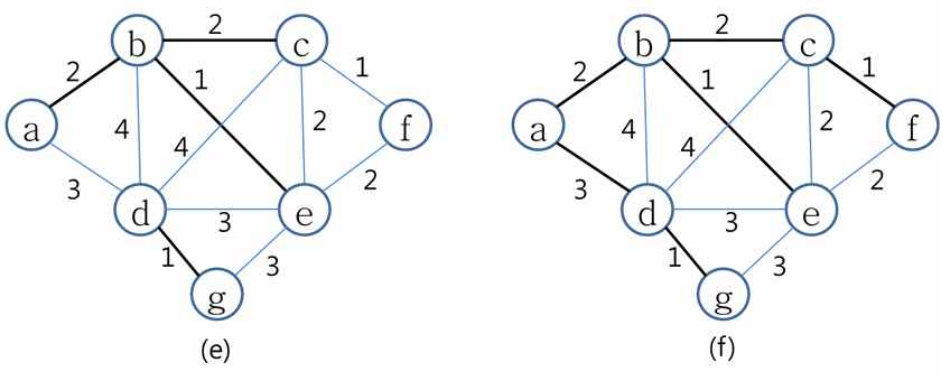
(b)



(c)



(d)



- 가중 그래프
- 가중치의 합은 10이다.

3. 크루스칼의 알고리즘의 개요

• 욕심쟁이 방법을 적용한 것
• 사이클을 만들지 않는 최단 간선을 하나씩 추가해 나감에 의해 최소 신장 나무를 구하는 방법으로 간선을 가중치가 증가하는 순으로 한번에 하나씩 사이클이 만들어지지 않도록 추가해 나감

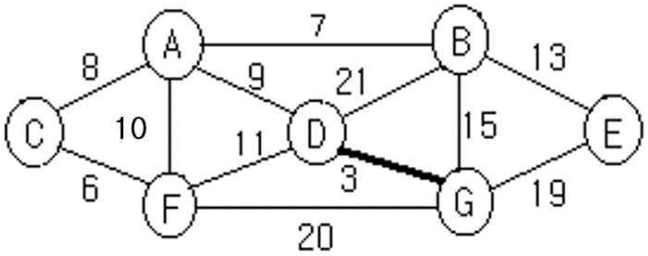
① 크루스칼의 알고리즘 탐색 과정
• 가중치가 주어진 그래프에서 최소 가중치를 갖는 간선부터 하나씩 선택
• 사이클이 형성되지 않으면 신장 트리 T에 포함
• 이 과정을 간선의 수가 n-1(정점의 개수 : n)개가 될 때까지 반복한다.
ⓐ 그래프에서 최소 가중치를 갖는 간선부터 차례대로 선택하여 최소 비용 신장 트리에 포함시킨다. 최소 비용 트리의 간선의 수가 n-1개이면 종료한다.
ⓑ 만약 선택된 간선이 사이클을 형성하면 최소 비용 신장 트리에 포함시키지 않고 위의 과정을 반복한다. Kruskal방법을 적용하여 주어진 가중치 그래프 G의 최소 비용신장 트리를 유도하는 과정
• 최소 비용의 간선을 선택하는 것은 우선순위 큐를 사용하거나, 간선들을 비용에 대해서 순서 배열함

② 크루스칼의 알고리즘

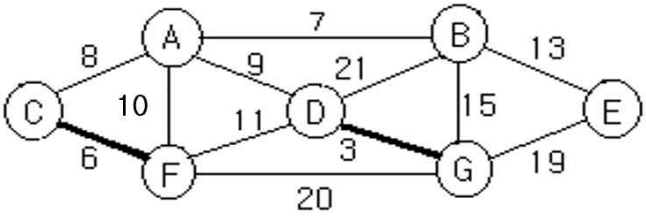
```
Kruskal (Adj(G), T) {
/* 입력: 그래프 G=(V,E)의 인접 리스트 출력: 최소 신장 나무의 간선 집합 T*/
T =     ϕ;
for ( 그래프 G의 각 정점 v)
init_S(v);
/* 가중치가 증가하는 순으로 E의 간선을 정렬 */
for(가중치가 작은 것부터 모든 간선 (u,v) ∈ E)
if (find(u) ≠ find(v)) {
T=T ∪ {u,v)}
union(u,v);
}
}
```

③ 가중치 그래프를 최소 비용 신장 트리로 만드는 과정

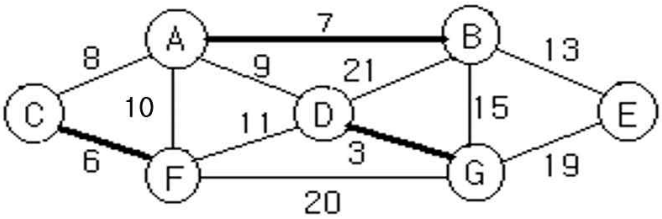
㉠ 가중치가 3인 연결선(D,G) 선택



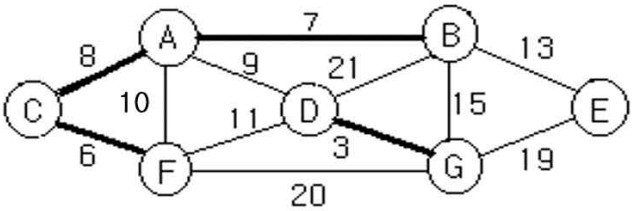
㉡ 가중치가 6인 연결선 (C,F) 선택



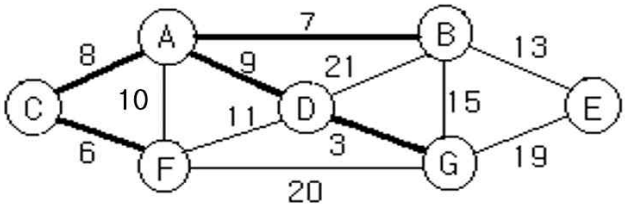
㉢ 가중치가 7인 연결선 (A,B) 선택



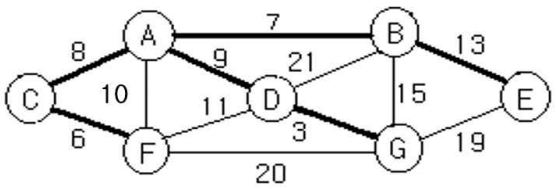
㉣ 가중치가 8인 연결선 (A,C) 선택



㉤ 가중치가 9인 연결선 (A,D) 선택

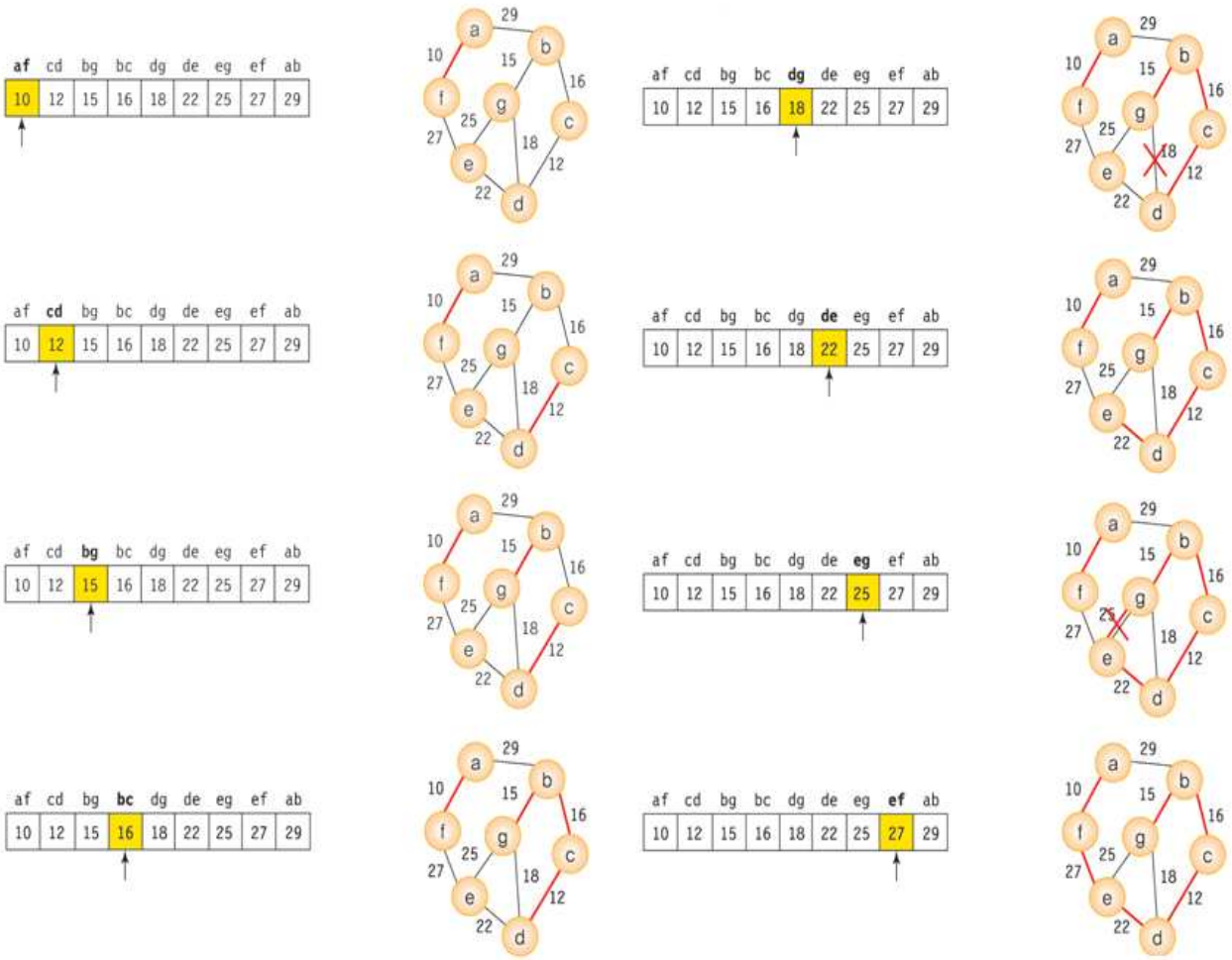


㉥ 가중치가 10인 연결선 (A,F)는 사이클을 형성하므로 배제 가중치가 11인 연결선 (D,F)는 사이클을 형성하므로 배제 가중치가 13인 연결선 (B,E) 선택



㉦ 가중치가 15인 연결선 (B,G)는 사이클을 형성하므로 배제가중치가 19인 연결선 (E,G)는 사이클을 형성하므로 배제 가중치가 20인 연결선 (F,G)는 사이클을 형성하므로 배제 또는, ㉥의 과정까지 선택된 연결선의 수가 이미 (n-1)개이므로 실행 종료

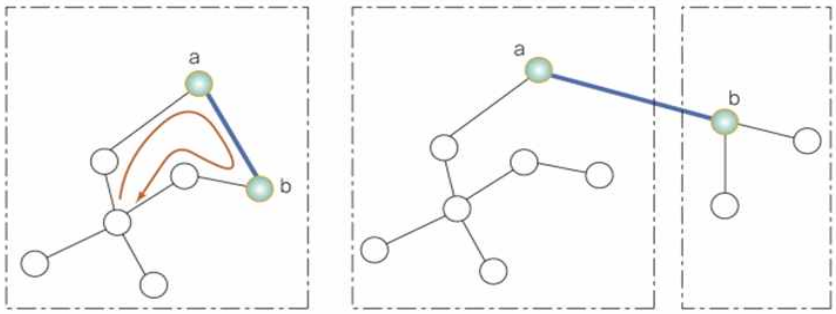
다른 예)



4. union-find 알고리즘(Kruskal의 알고리즘에서 합-찾기)

◎ 두 집합들의 합집합 만듬

- ☞ 원소가 어떤 집합에 속하는지 알아냄
- ☞ Kruskal의 MST 알고리즘에서 사이클 검사에 사용



(a) 사이클 형성 (b) 사이클 형성되지 않음

5. Kruskal 알고리즘의 시간 복잡도

⦿ Kruskal 알고리즘은 대부분 간선들을 정렬하는 시간에 좌우됨
☞ 사이클 테스트 등의 작업은 정렬에 비해 매우 신속하게 수행됨

⦿ 네트워크의 간선 e개를 퀵정렬과 같은 효율적인 알고리즘으로 정렬한다면 Kruskal 알고리즘의 시간 복잡도는 O(e*log(e))가 된다

학습내용2 : 프림 알고리즘

• 가중치가 주어진 그래프에서 임의의 정점을 선택하여 → 이 정점에 연결된 간선들 중 가중치가 제일 작은 간선을 선택하여 → 최소 비용 신장 트리에 포함
• 그 후부터는 최소 비용 신장 트리에 포함된 간선들에 인접한 정점들에 연결된 간선들에 대해 가중치가 제일 작은 간선을 선택하여 → 사이클이 형성되지 않으면 최소 비용 신장 트리에 포함 .
• 이 과정을 간선의 수가 n-1개가 될 때까지 반복

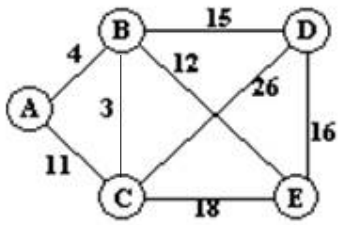
1. 프림 알고리즘의 수행과정

① 시작할 정점을 임의로 선택한다.
② 선택된 정점들에 부속된 간선들 중 가중치가 제일 작은 간선을 선택하여 최소비용신장 트리에 포함시킨다.
③ 선택된 간선에 인접한 정점들에 부속된 간선들에 대해서도 ⓑ를 반복한다.
④ 만약 사이클이 발생하면 선택한 간선은 제거하고 그 다음 가중치가 작은 간선이 있으 면 선택 하고 없으면 그 전에 방문한 정점에 연결된 간선에 대해 ⓑ를 반복한다.
⑤ 모든 정점이 연결되면 종료한다.

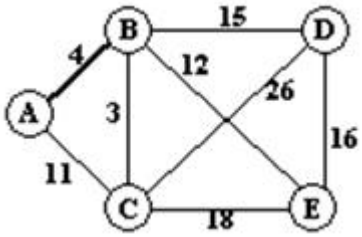
2. 알고리즘

```
Prim(G, T){
/* 입력 : 그래프 G=(V,E) 출력: 최소 신장 나무의 간선 집합 */
{
T =     ϕ;
S = {1}; /* 정점들의 집합-첫번째 정점으로 초기화 */
while (S!= V) {
u ∈S, v ∈ V-S인 것 중 최소인 간선 (u,v) 선택;
T=T ∪ {(u,v)};
S=S ∪ {v};
}
}
```

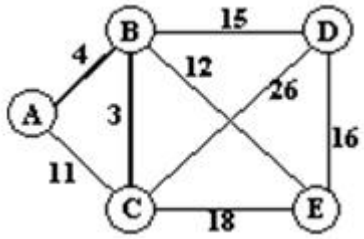
3. 가중치 그래프를 최소 비용 신장 트리로 만드는 과정



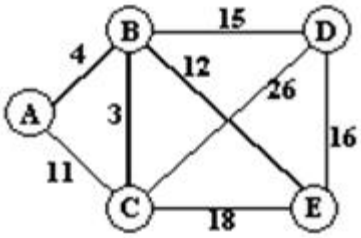
ⓐ 정점A를 선택, A에서 연결된 간선중 가중치가 제일작은 4를 선택



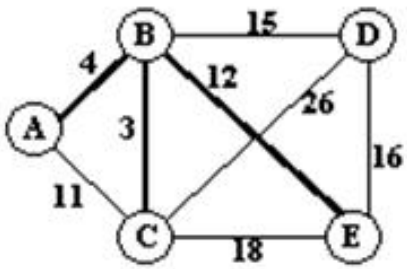
ⓑ 정점 A,B에 연결된 간선 중 가중치가 제일작은 3 선택



ⓒ 정점 A,B,C에 연결된 간선중 최소 가중치를 갖는 12를 선택

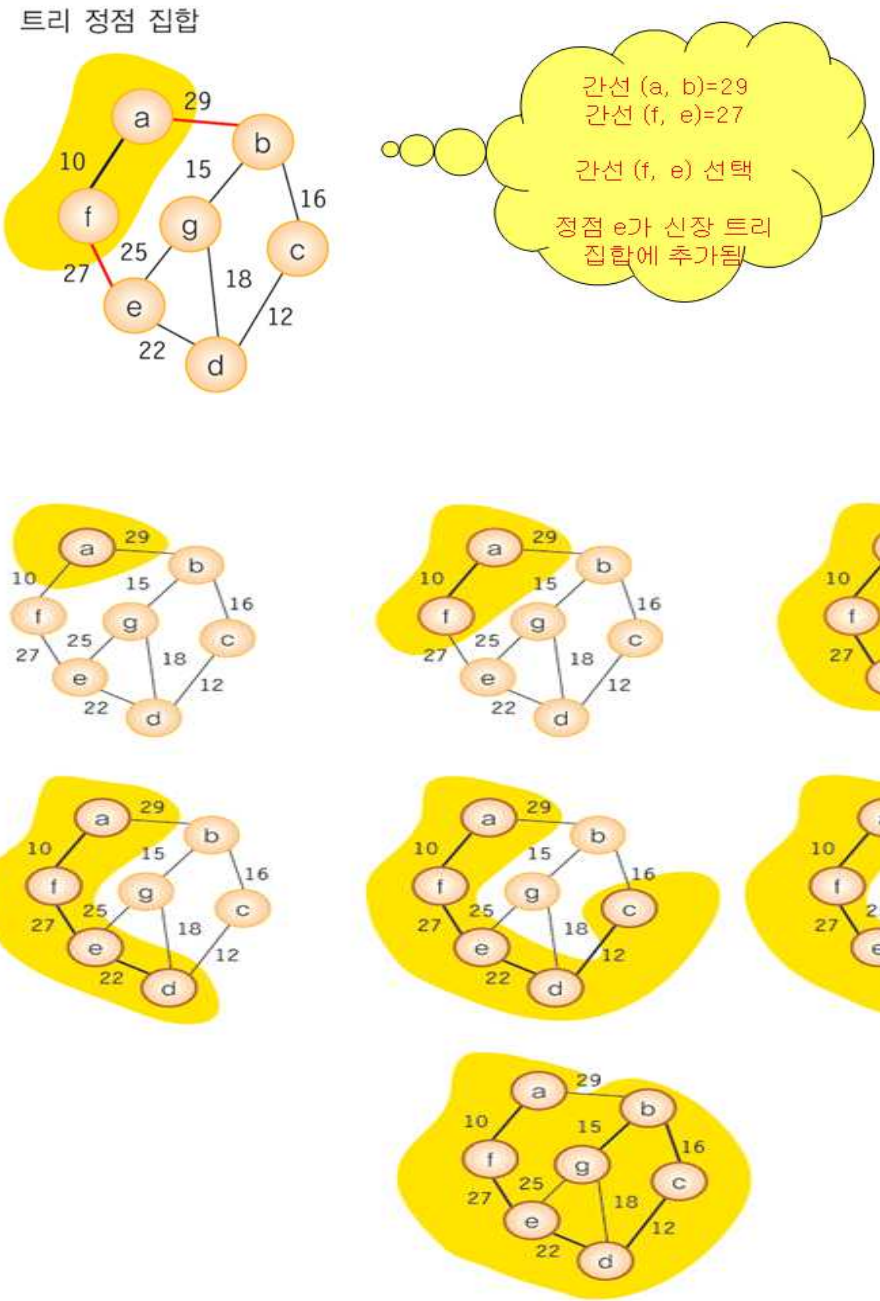


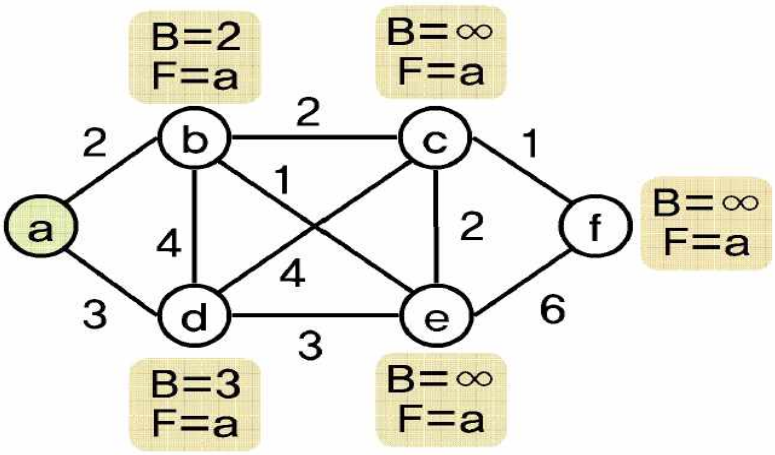
ⓓ 정점 A,B,C,E에 연결된 간선중 가중치가 제일작은 15선택



4. 프림의 알고리즘

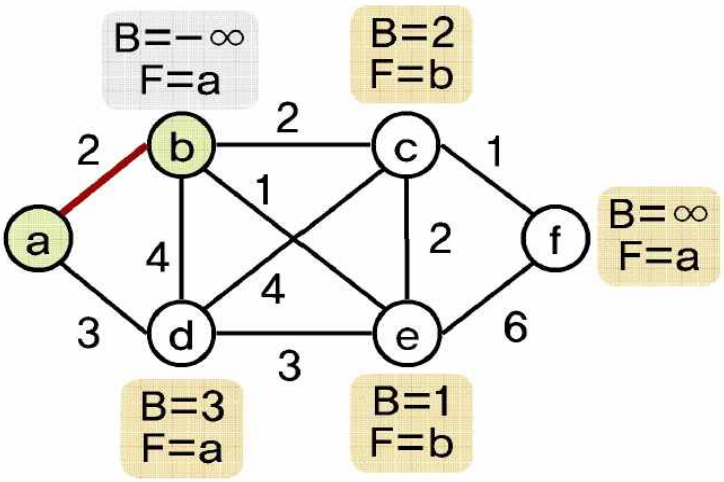
◎ 시작 정점에서부터 출발하여 신장 트리 집합을 단계적으로 확장해나감
  ☞ 시작 단계에서는 시작 정점만이 신장 트리 집합에 포함됨
◎ 신장 트리 집합에 인접한 정점 중에서 최저 간선으로 연결된 정점 선택하여 신장 트리 집합에 추가함
◎ 이 과정은 신장 트리 집합이 n-1개의 간선을 가질 때까지 반복

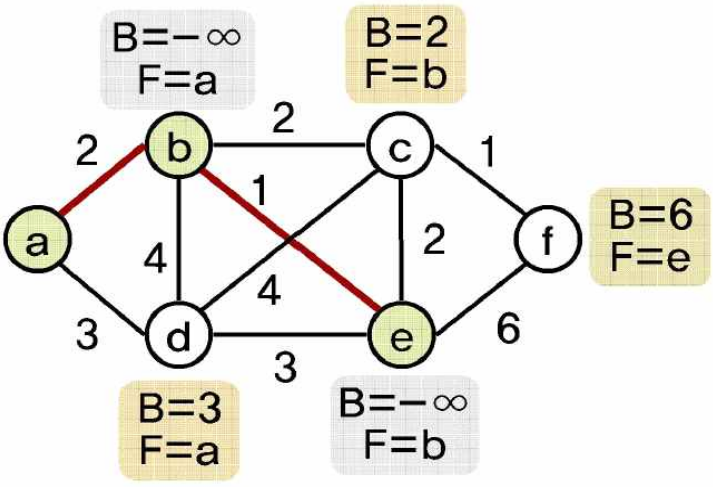


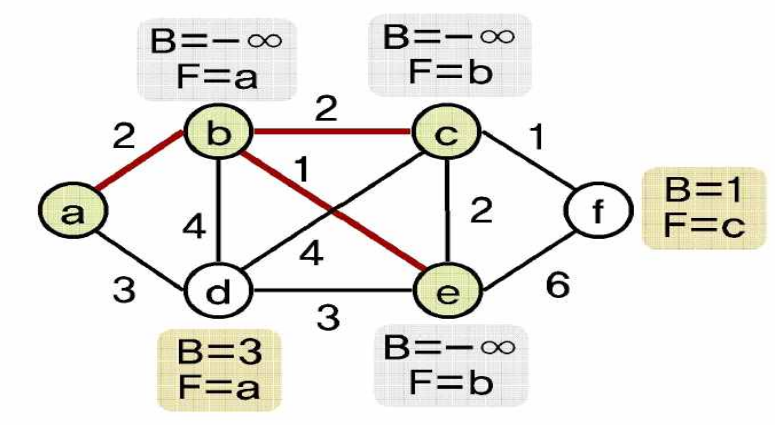


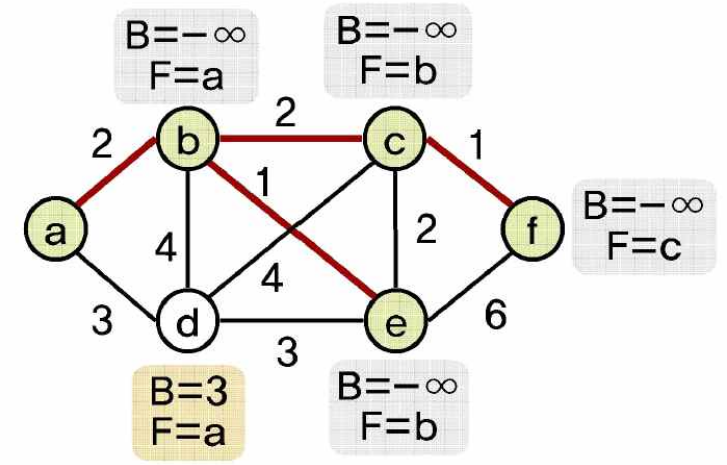
BEST와 FROM에 의한 최소 간선의 선택

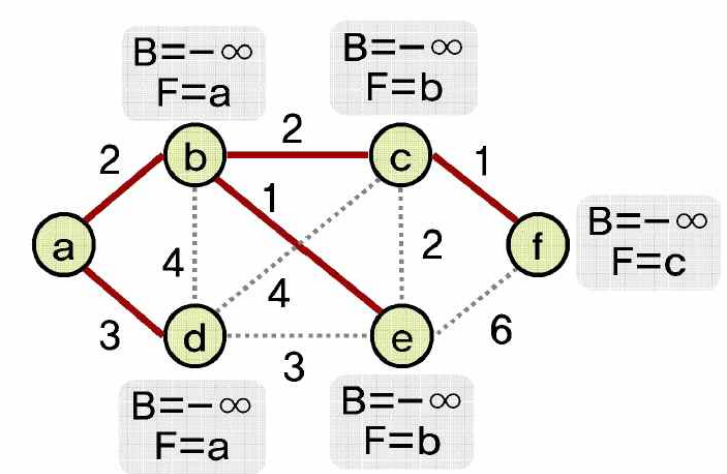
BEST[i] : S와 정점 i를 잇는 간선의 가중치 중 최소값
FROM[i] : BEST[i]인 간선의 다른 끝 정점 번호











◎ 프림 알고리즘의 시간 복잡도

☞ 시간 복잡도: 인접행렬 - $O(|V|^2)$

☞ 인접리스트 - $O((|V|+|E|)\lg|V|)$

【학습정리】

1. 최소 신장 나무

- 최소 신장 나무(MST, minimum spanning tree)란 신장 나무 중에서 간선의 가중치 합이 작은 것
  → 신장 나무 : 주어진 그래프의 모든 노드들을 포함하는 나무
- 크루스칼 방법과 프림의 방법 → 두 방법 모두 욕심쟁이 방법이 적용됨
- 크루스칼 알고리즘
  → 간선이 하나도 없는 숲에서 시작하여 사이클을 만들지 않는 최소 간선들을 하나씩 추가해 나가는 방법
  → 알고리즘에서 사이클의 존재 여부를 조사하기 위해 합-찾기 연산이 사용됨
  → 시간 복잡도 O(|E|lg|E|)
- 프림 알고리즘
  → 이미 선택된 정점에 부수된 최소 간선을 추가해 나가는 방법 = 이미 선택된 정점 집합 S와 V-S를 잇는 최소의 간선을 선택해서 추가하는 방법
  → 시간 복잡도: 인접행렬 - $O(|V|^2)$, 인접리스트 - O((|V|+|E|)lg|V|)