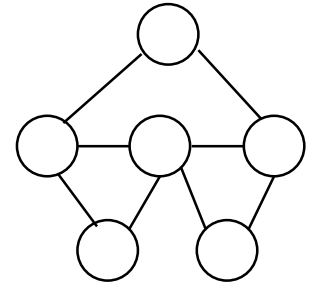


그래프1(복습)

1. 다음과 같이 정점(vertex)의 집합 V 와 연결선(edge)의 집합 E 로 구성된 그래프 G 에 대하여 다음 물음에 답하시오.

$$V(G) = \{1, 2, 3, 4, 5, 6\}$$

$$E(G) = \{(1,2), (1,4), (2,3), (2,5), (3,4), (3,5), (3,6), (4,6)\}$$



(1) 해당 그래프 G 를 그리시오.

(2) 위 그래프에 해당하는 인접행렬(adjacency matrix)과 인접리스트(adjacency list)를 그리시오.
(인접리스트표현의 경우 각 정점의 연결리스트는 정점을 표시하는 숫자순으로 연결되어 있다고 가정하자.)

| | | | | | |
|--|--|--|--|--|--|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

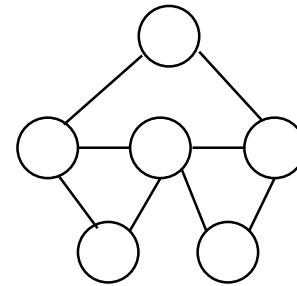
| | | | | |
|--|--|--|--|--|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

그래프1(복습)

(3) 다음의 방법으로 탐색한 순서를 정점 번호로 쓰시오.

(3-1) 깊이우선탐색(Depth First Search) dfs(1)

(3-2) 너비우선탐색(Breadth First Search) bfs(1)



그래프의 응용

◆ 연결요소(connected component) 구하기

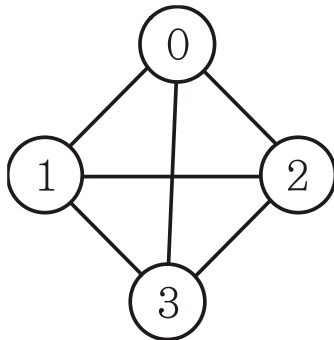
- 방문하지 않은 정점 v 에 대해 $DFS(v)$ 또는 $BFS(v)$ 를 반복 호출로 구함
- $dfs(0)$ or $bfs(0)$ 를 호출한 후 방문이 안된 정점이 있는 경우

```
int connected()
{
    int v, cnum=0;
    for(v=0; v<n; v++)
        if(already_visited[v] == 0) {
            dfs(v);
            cnum++;
            printf( "\n" );
        }
    return cnum;
}
```

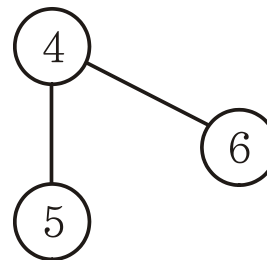
그래프의 응용

◆ 연결요소(connected component)

- $V(G) = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$
- $E(G) = \{(0,1), (0,2), (0,3), (1,2), (1,3), (2,3), (4,5), (4,6), (7,8)\}$



C1



C2



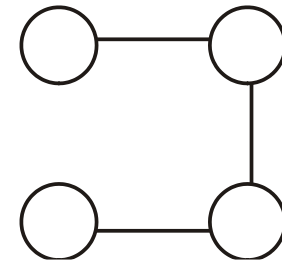
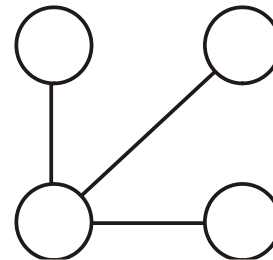
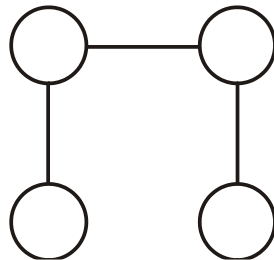
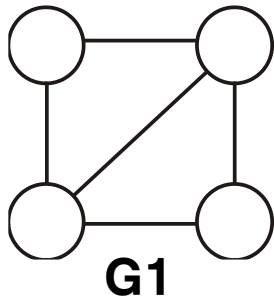
C3

3개의 연결요소로 표현 가능

신장 트리(spanning tree)

◆ n 개의 정점으로 이루어진 무방향 그래프 G 에서
 n 개의 모든 정점과 $n-1$ 개의 간선으로 만들어진 트리

◆ 그래프 G_1 과 신장 트리의 예



신장 트리(spanning tree)

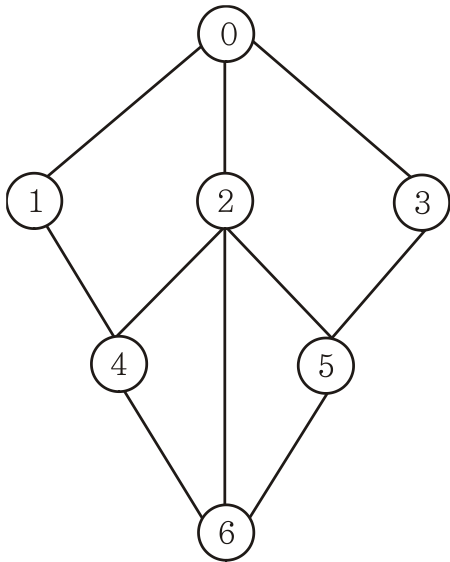
◆깊이 우선 신장 트리(depth first spanning tree)

- 깊이 우선 탐색을 이용하여 생성된 신장 트리

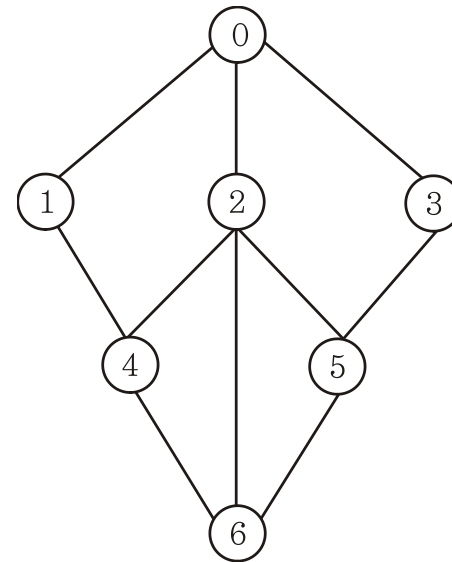
◆너비 우선 신장 트리(breadth first spanning tree)

- 너비 우선 탐색을 이용하여 생성된 신장 트리

신장 트리(spanning tree)

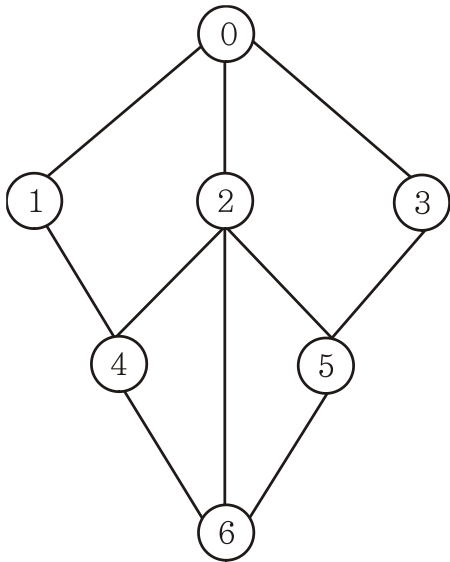


(a) dfs(4) 신장 트리

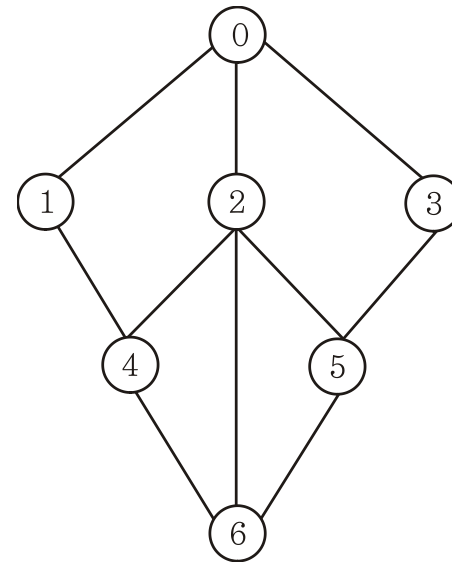


(b) bfs(4) 신장 트리

신장 트리(spanning tree)



(a) dfs(0) 신장 트리



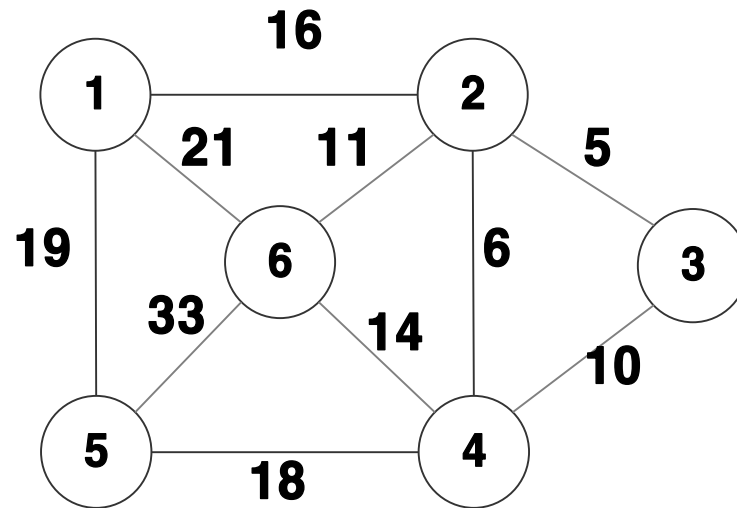
(b) bfs(0) 신장 트리

그래프의 응용

◆ 최소 비용 신장 트리(minimum cost spanning tree)

- 무방향 가중치 그래프에서 신장 트리를 구성하는 간선들의 가중치 합이 최소인 신장 트리
 - 가중치 그래프의 간선에 주어진 가중치
 - ✓ 비용이나 거리, 시간을 의미하는 값
- 최소 비용 신장 트리를 만드는 알고리즘
 - Kruscal 알고리즘

그래프의 응용



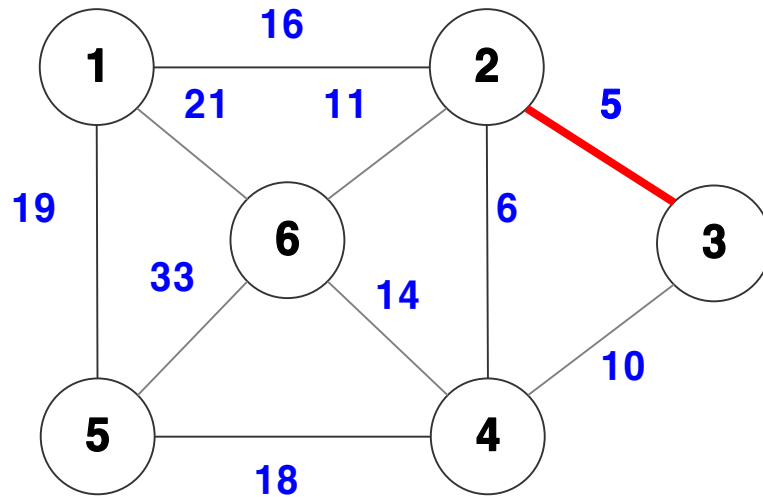
[가중치가 있는 그래프]

최소 비용 신장 트리

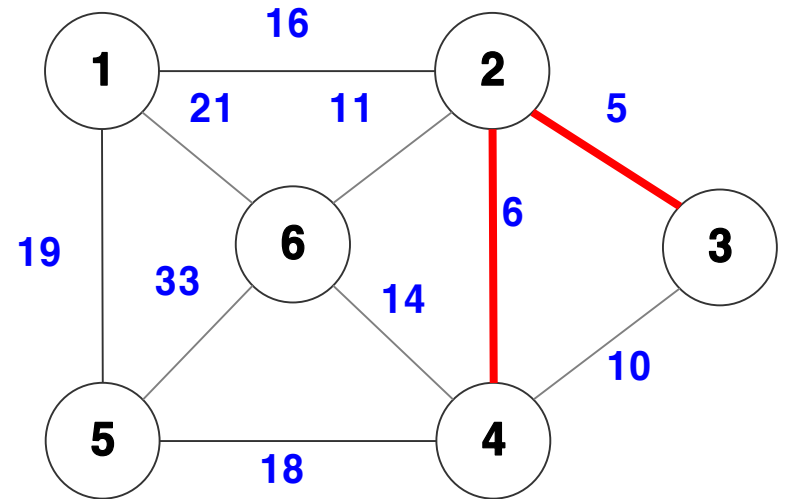
◆ Kruscal 알고리즘

- 간선중 가장 가중치가 적은 간선의 순으로 선택
- 사이클이 발생하면 제외.
- 과정 반복 후, 모든 정점이 연결되면 알고리즘 끝낸다.

최소 비용 신장 트리

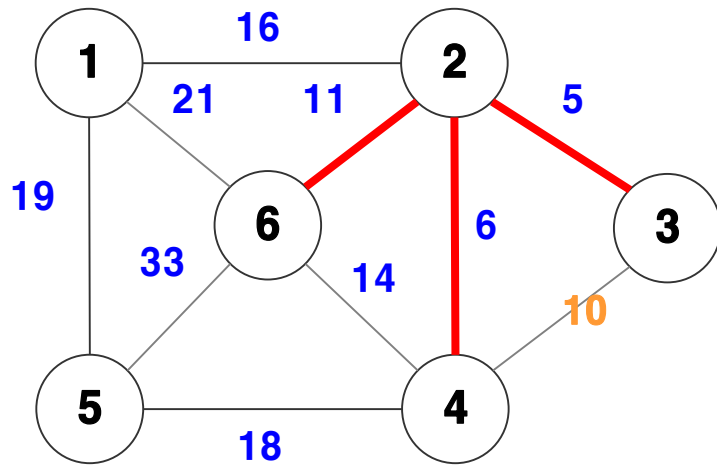


① 가장 적은 가중치 5를 선택

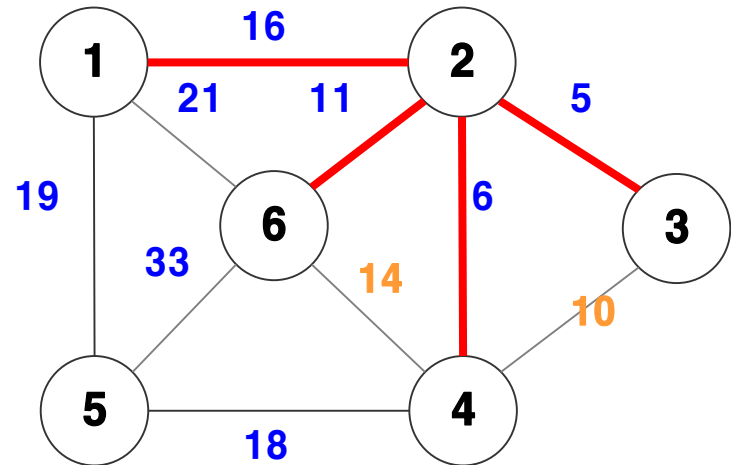


② 두번째 적은 가중치 6을 선택

최소 비용 신장 트리

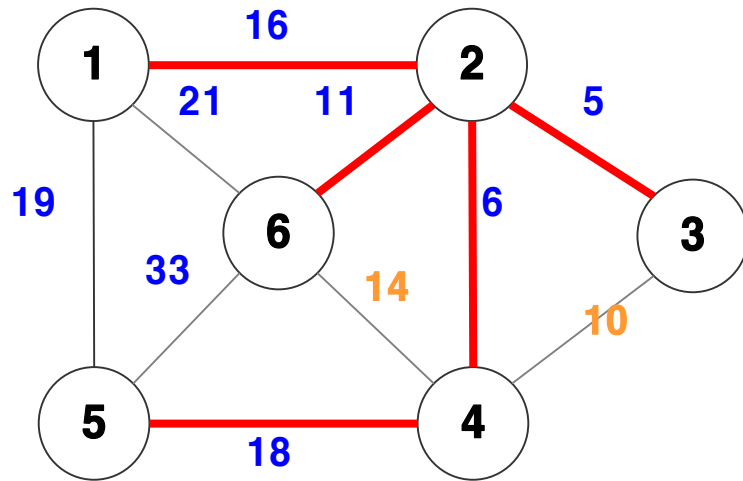


- ③ 세번째 적은 가중치 10을 선택하면 Cycle이 발생하여, 제외 네번째 적은 11을 선택

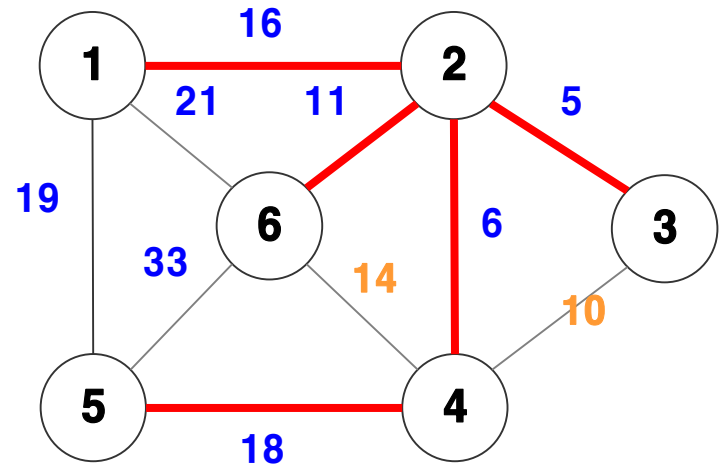


- ④ 다섯번째 적은 14를 선택하면 Cycle이 발생하여 제외, 여섯번째 적은 16을 선택

최소 비용 신장 트리



⑤ 일곱번째 적은 18을 선택



⑥ 모든 정점이 연결된 신장 트리가 완성

$$\text{총비용} = 5 + 6 + 11 + 16 + 18 = 56$$