

GLAPH

Chapter 9

Sogang ACM-ICPC Team

Heewon Yun

1/31 2013

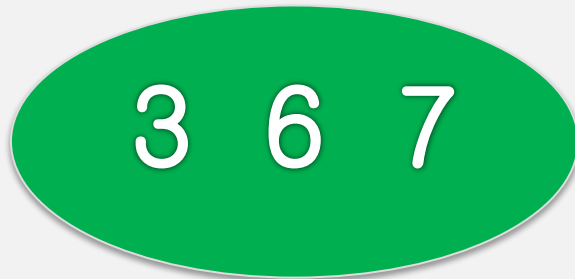
강의순서

- ✓ Union-Find Tree
- ✓ Minimum Spanning Tree
 - Kruscal algorithm
 - Prim algorithm

Union-Find tree

- 그룹으로 나누어 관리하는 데이터 구조
- 병합 가능! 분할 불가능!
- A와 B가 같은 그룹인지 검사(find)
- A와 B의 그룹 병합(union)

Union-Find tree

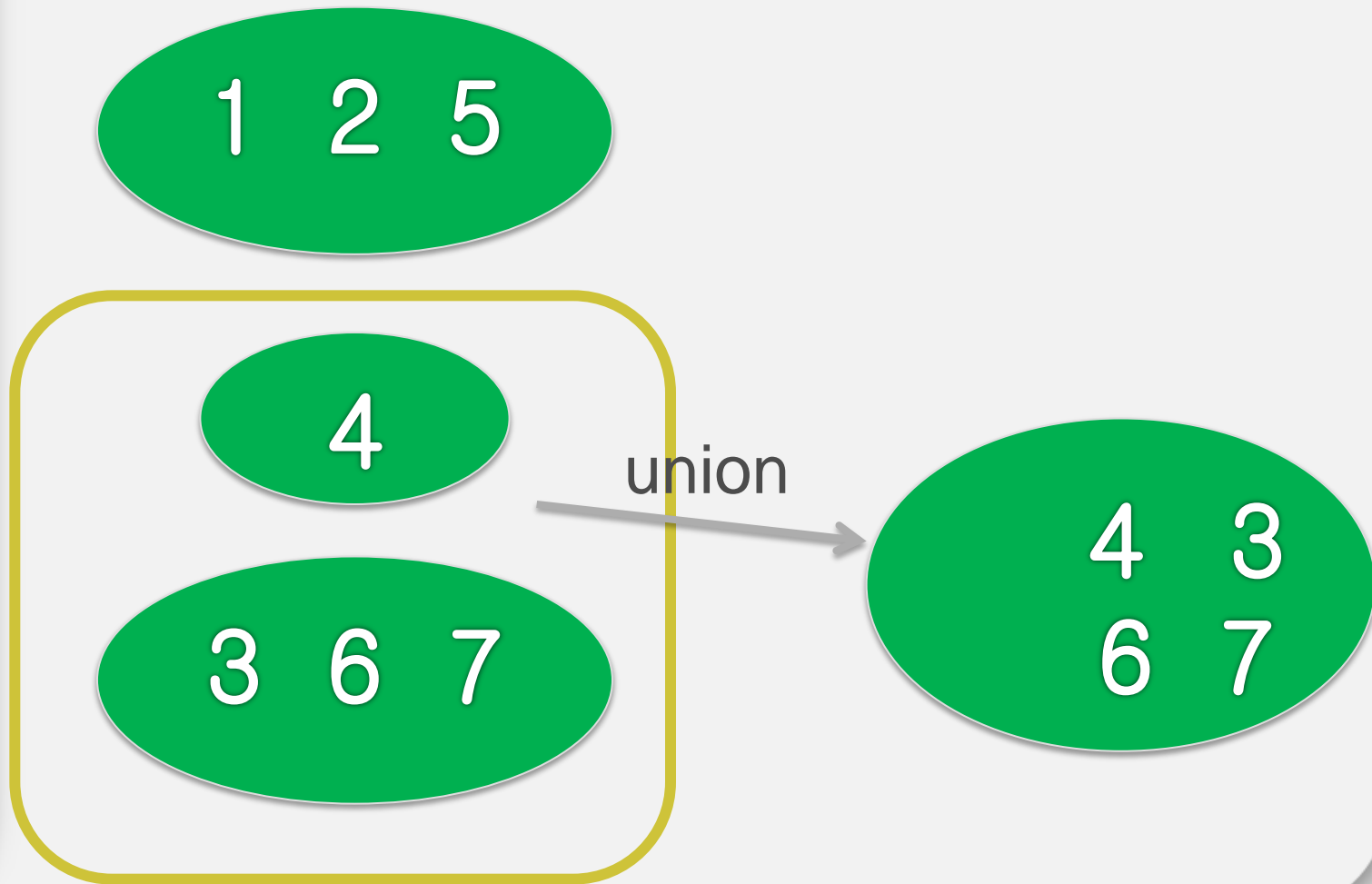


find



1과 5는 같은 그룹
1과 4는 다른 그룹

Union-Find tree



Union-Find tree

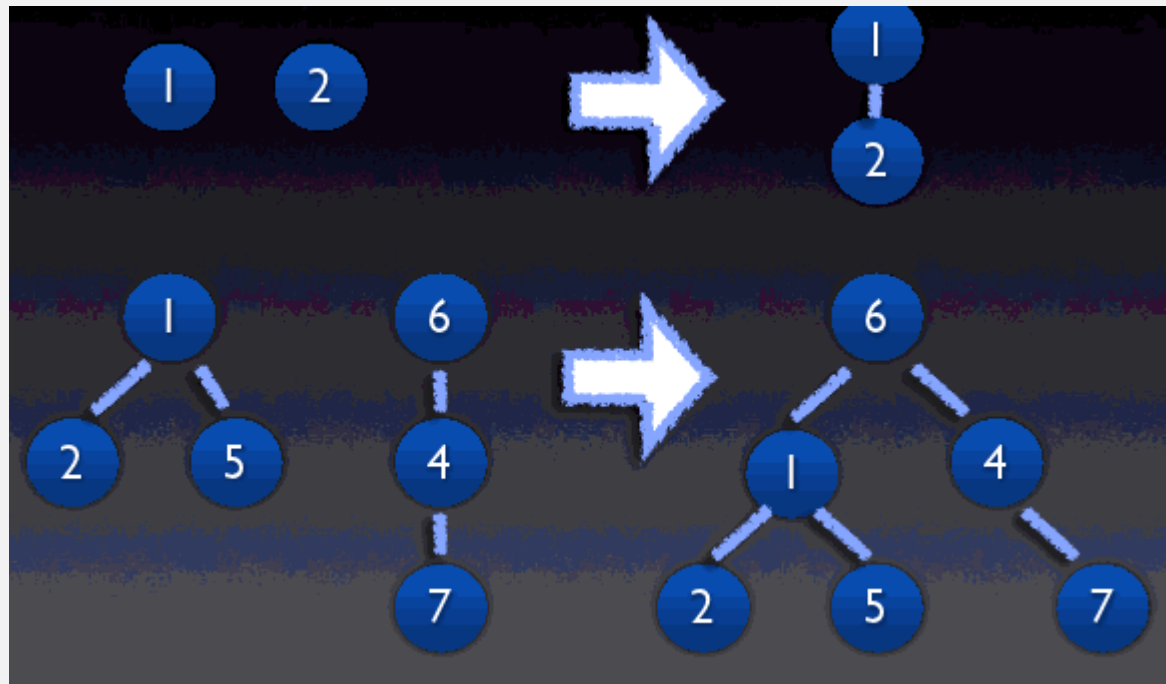
- 하나의 그룹은 하나의 트리!
- 부모 자식 관계는 의미 X
- 연결성만 중요!

Union-Find tree (initialize)

- N개의 요소에 대해 n개 노드 준비
- Edge는 아직 없는 상태
- 즉, 어떤 그룹에도 속하지 않는 상태

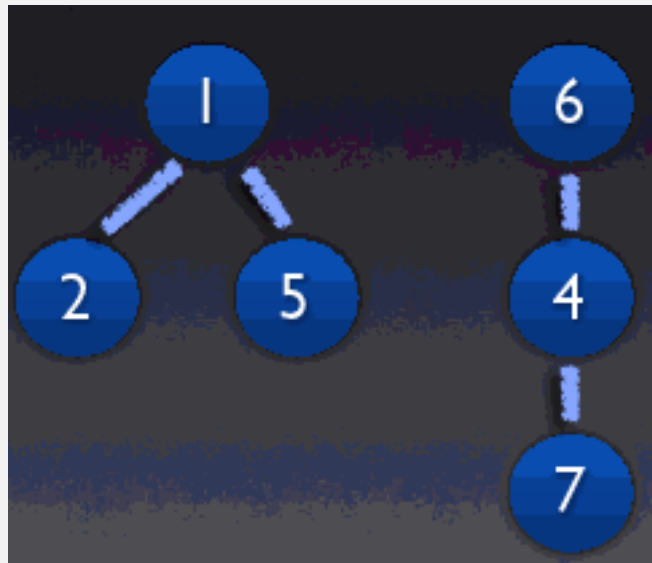
Union-Find tree (union)

- 루트로부터 다른 트리의 루트에 선을 연결



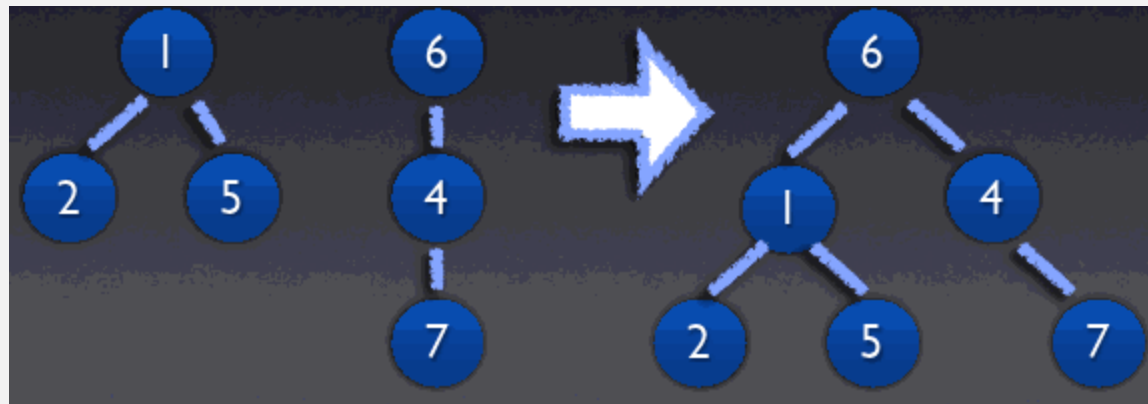
Union-Find tree (find)

- 같은 루트를 가지면 같은 그룹



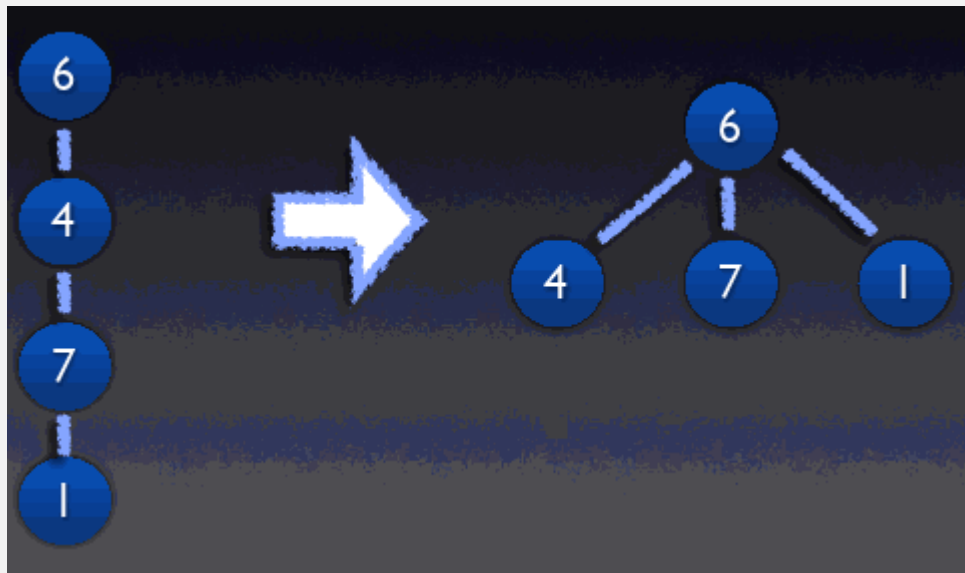
Union-Find tree (Rank)

- 트리가 한쪽에 편향되지 않게 해야함
- 각 트리에 대해 높이(Rank) 기억
- 병합시 Rank가 다르면 작은쪽이 큰쪽으로 들어감



Union-Find tree (Optimize)

- 탐색한 모든 노드를 루트 노드와 연결



Union-Find tree (Computation)

- 앞의 개념을 모두 적용시 매우 빠름!
- $O(n \log n)$ 보다 빠를 수 있다!

Union-Find tree (Implementation)

```
#define MAX_N 1111

int par[MAX_N]; // parent
int rank[MAX_N]; // depth of tree

//init
void init(int n){
    for(int i=0; i<n; i++){
        par[i] = i;
        rank[i] = 0;
    }
}
```

Union-Find tree (Implementation)

```
//get root
int find(int x){
    if(par[x] == x)
        return x;
    else
        return par[x] = find(par[x]);
}
```

Union-Find tree (Implementation)

```
//unite x & y
void union(int x, int y){
    x = find(x);
    y = find(y);
    if( x== y) return;
    if(rank[x] < rank[y])
        par[x] = y;
    else{
        par[y] = x;
        if(rank[x] == rank[y]) rank[x]++;
    }
}
```

Union-Find tree 관련문제

1717

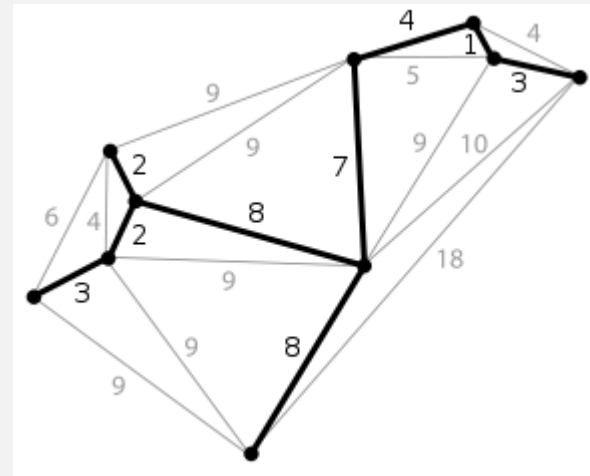
집합의 표현

Spanning tree

- 그래프의 모든 정점과 간선의 부분 집합으로 구성된 트리
- 무향 그래프
- 유일하지 않다

*스패닝 트리 중
가중치의 합이
가장 작은 트리

→ Minimum Spanning Tree



스패닝 트리의 예

MST

HOW?

MST

가중치가 작은 간선 선택!

POINT! Cycle이 없어야 한다.



크루스칼 알고리즘

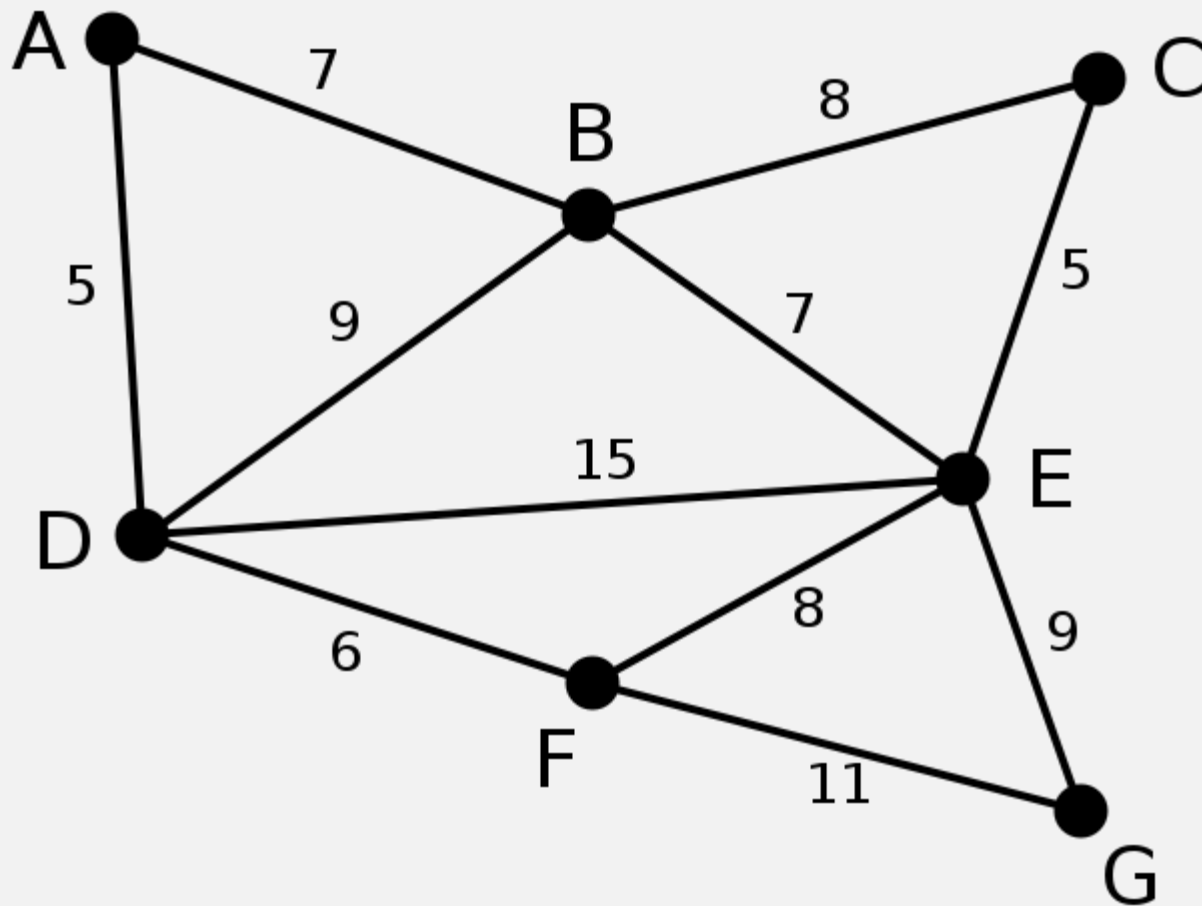


프림 알고리즘

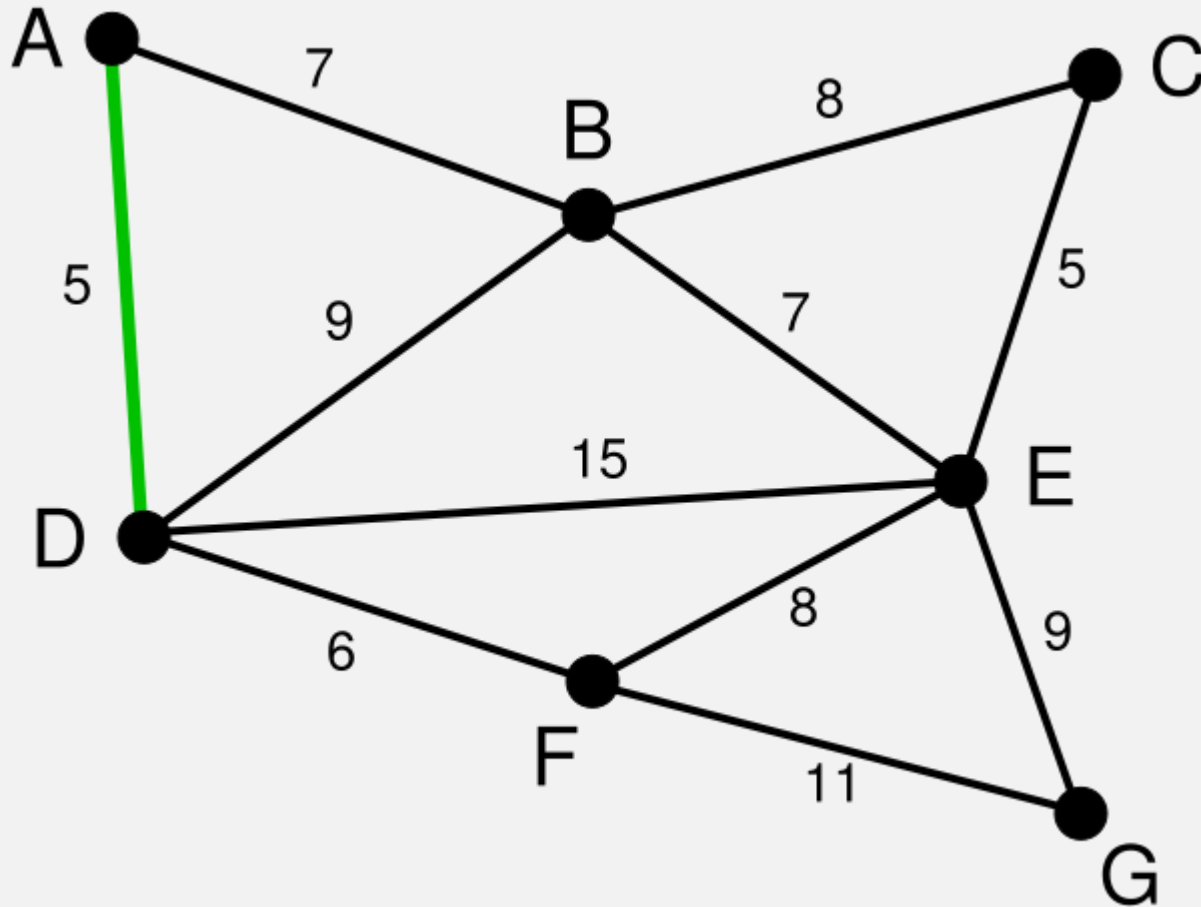
Kruscal algorithm

- ① 그래프의 모든 간선을 가중치를 기준으로 오름차순으로 정렬
- ② 가중치가 작은 간선 선택
- ③ 사이클이 생기지 않을 경우 스패닝 트리에 추가
- ④ 모든 간선을 한번씩 검사 후 종료

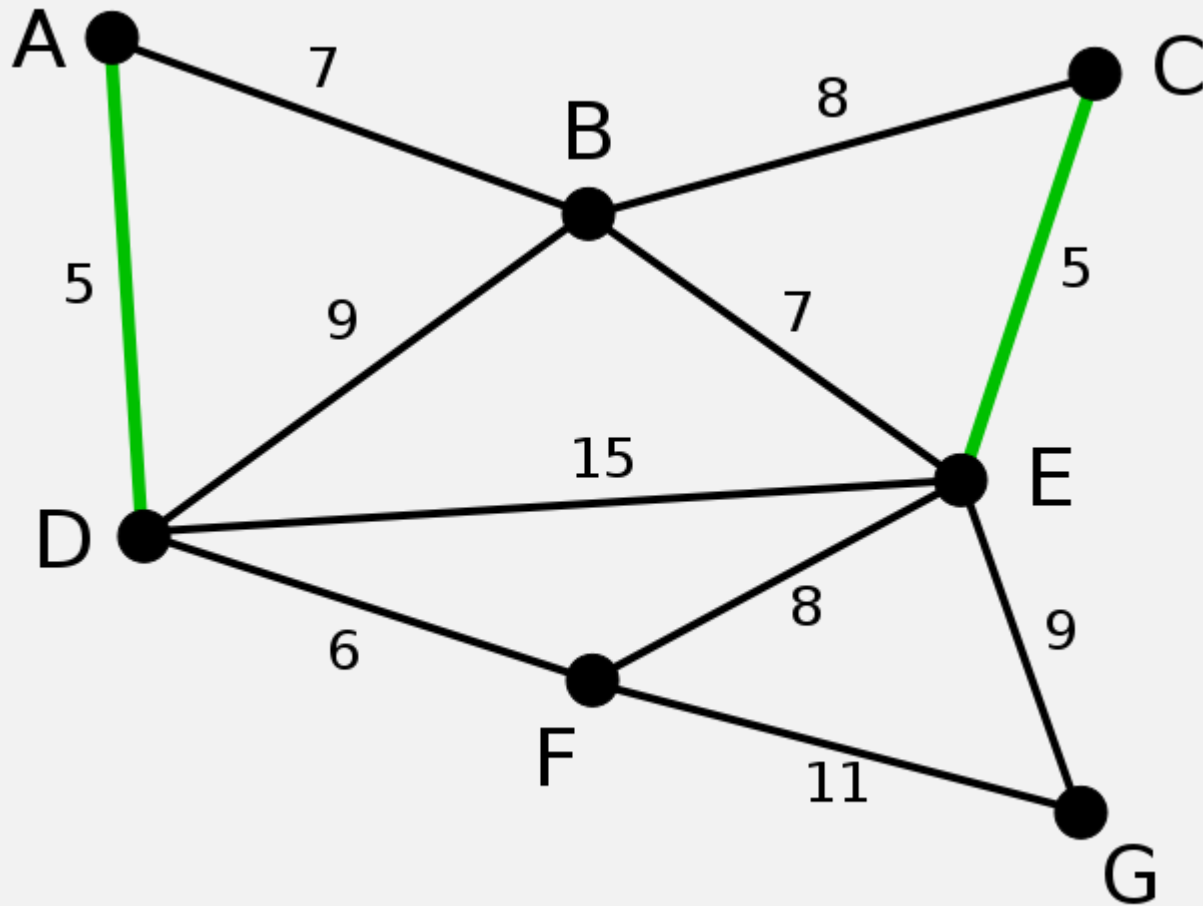
Kruskal algorithm



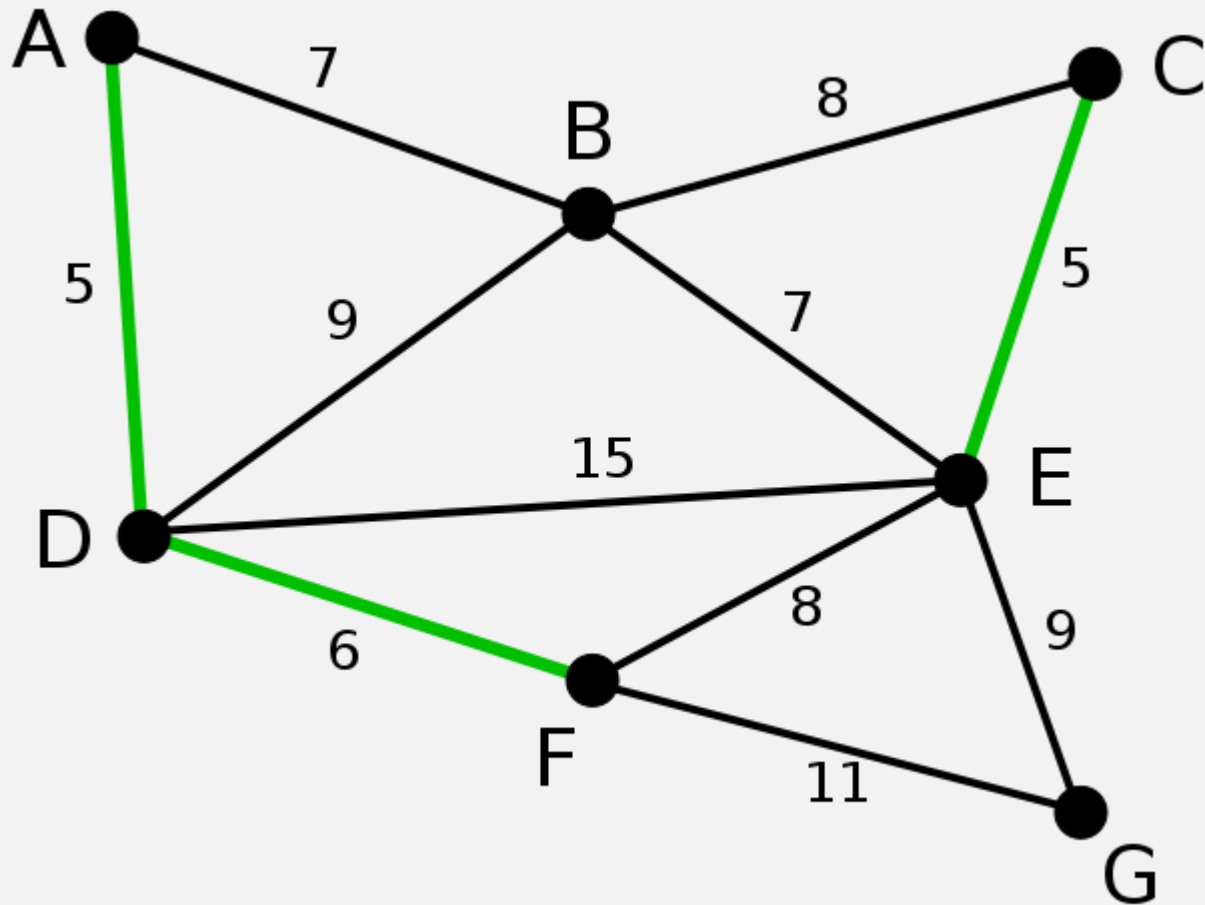
Kruskal algorithm



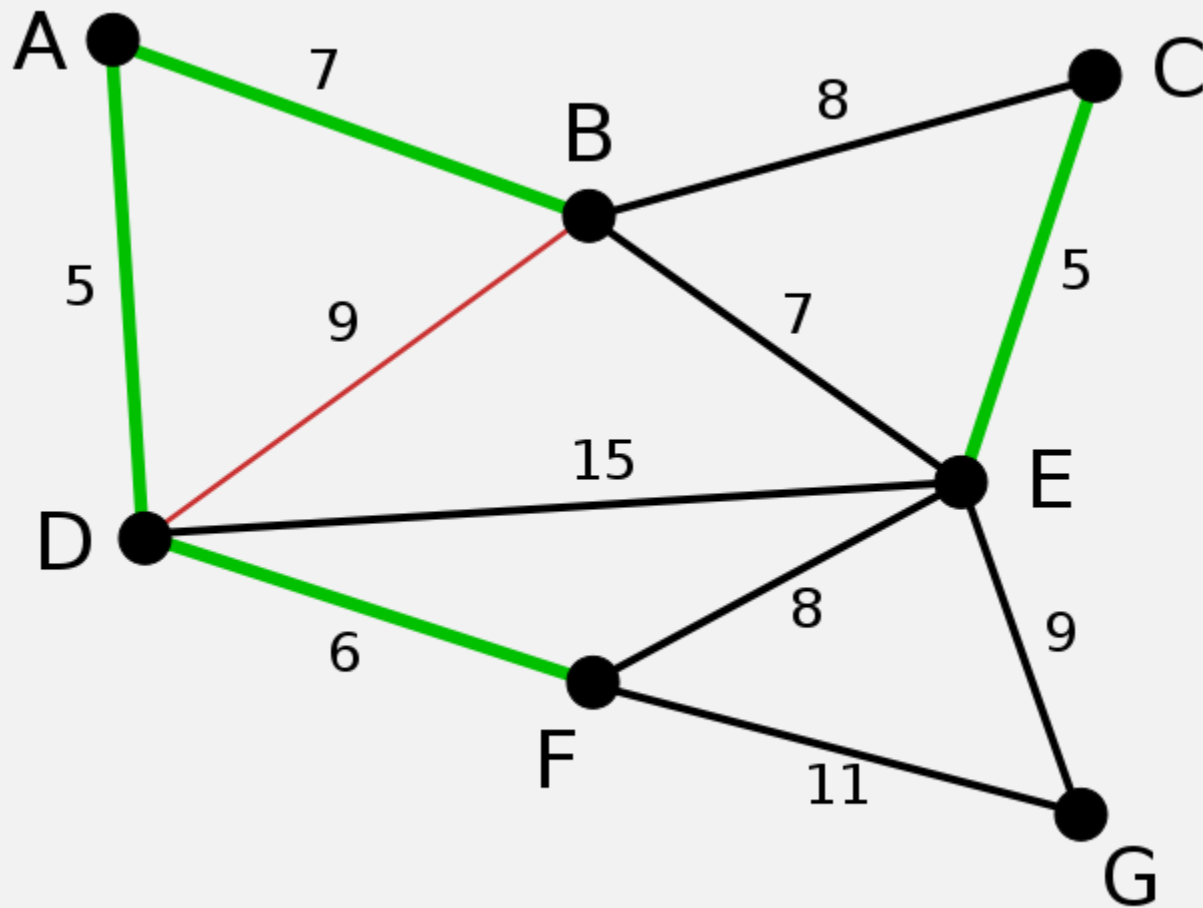
Kruskal algorithm



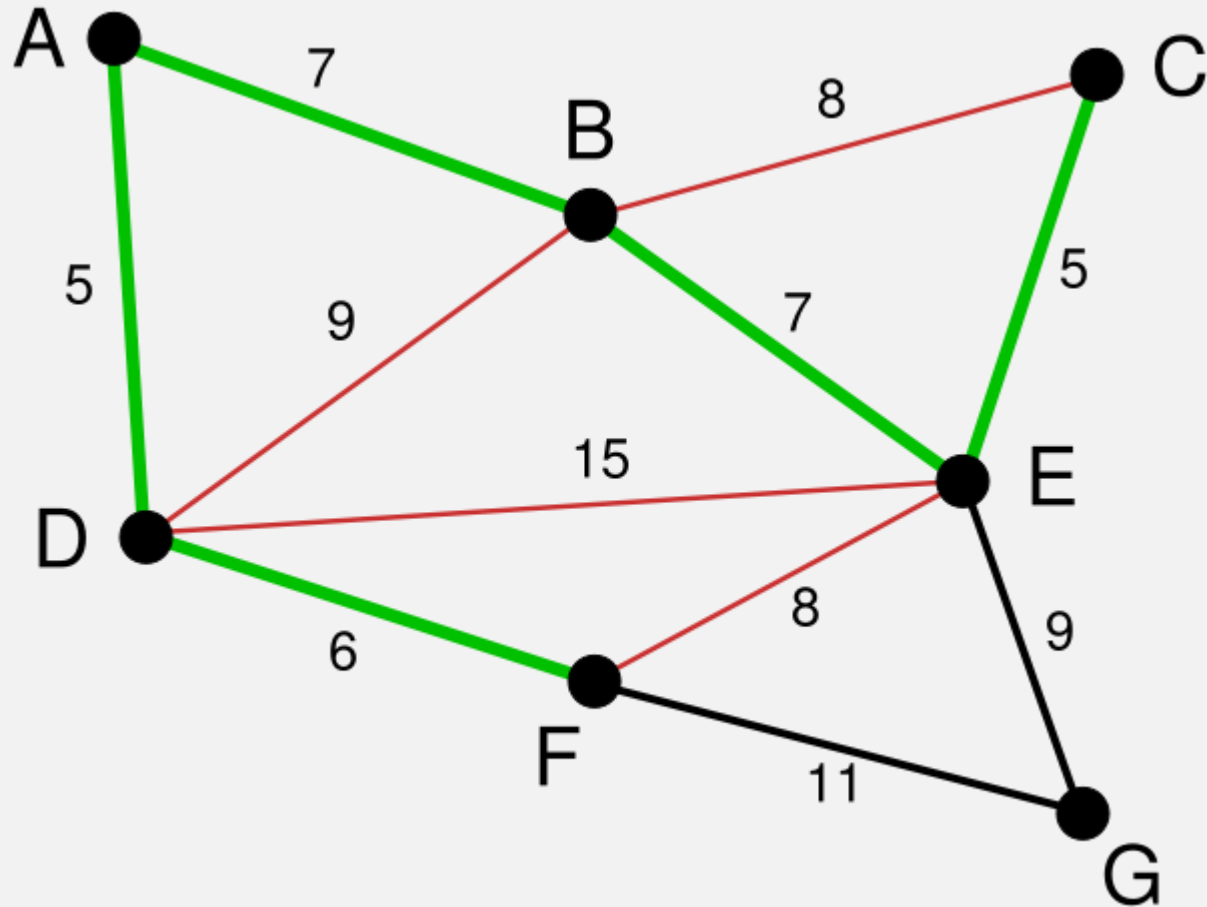
Kruskal algorithm



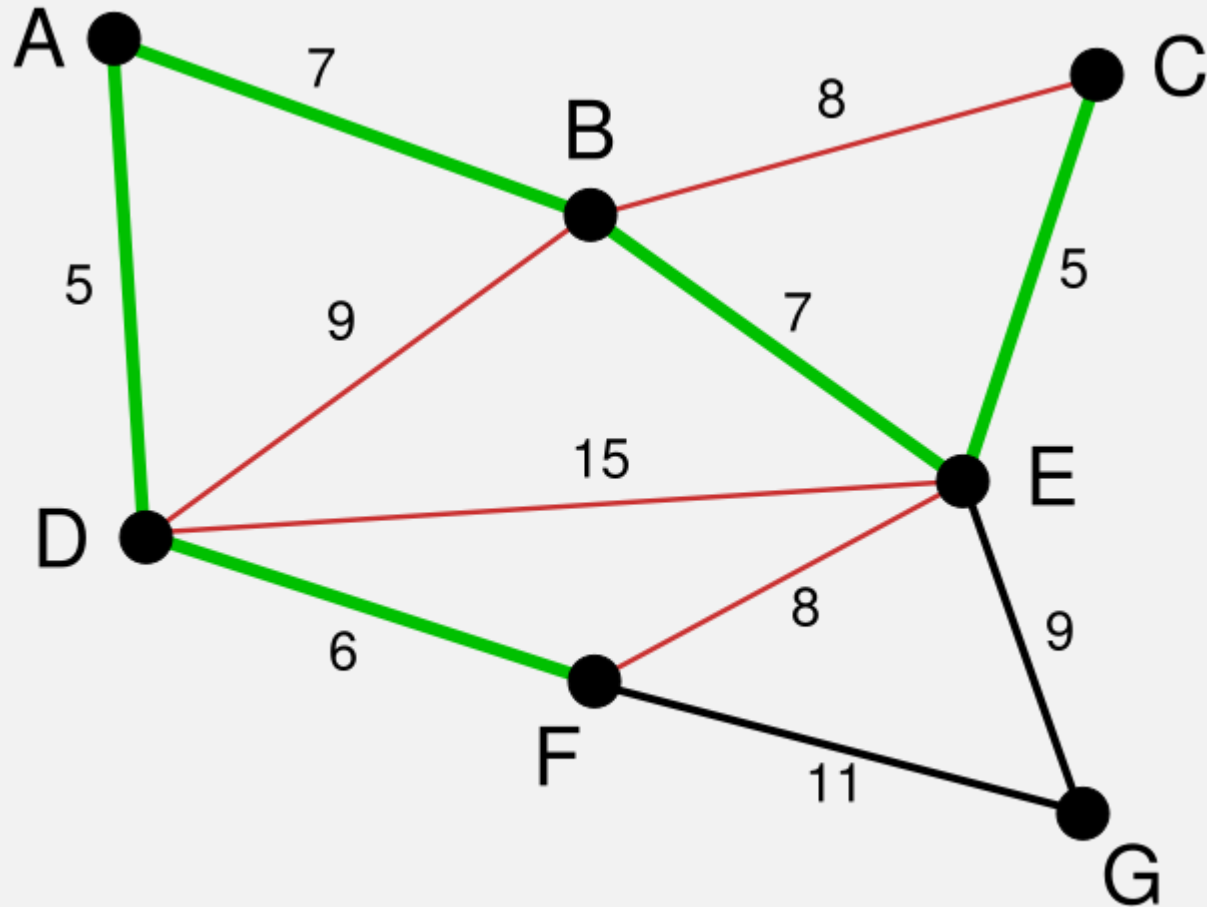
Kruskal algorithm



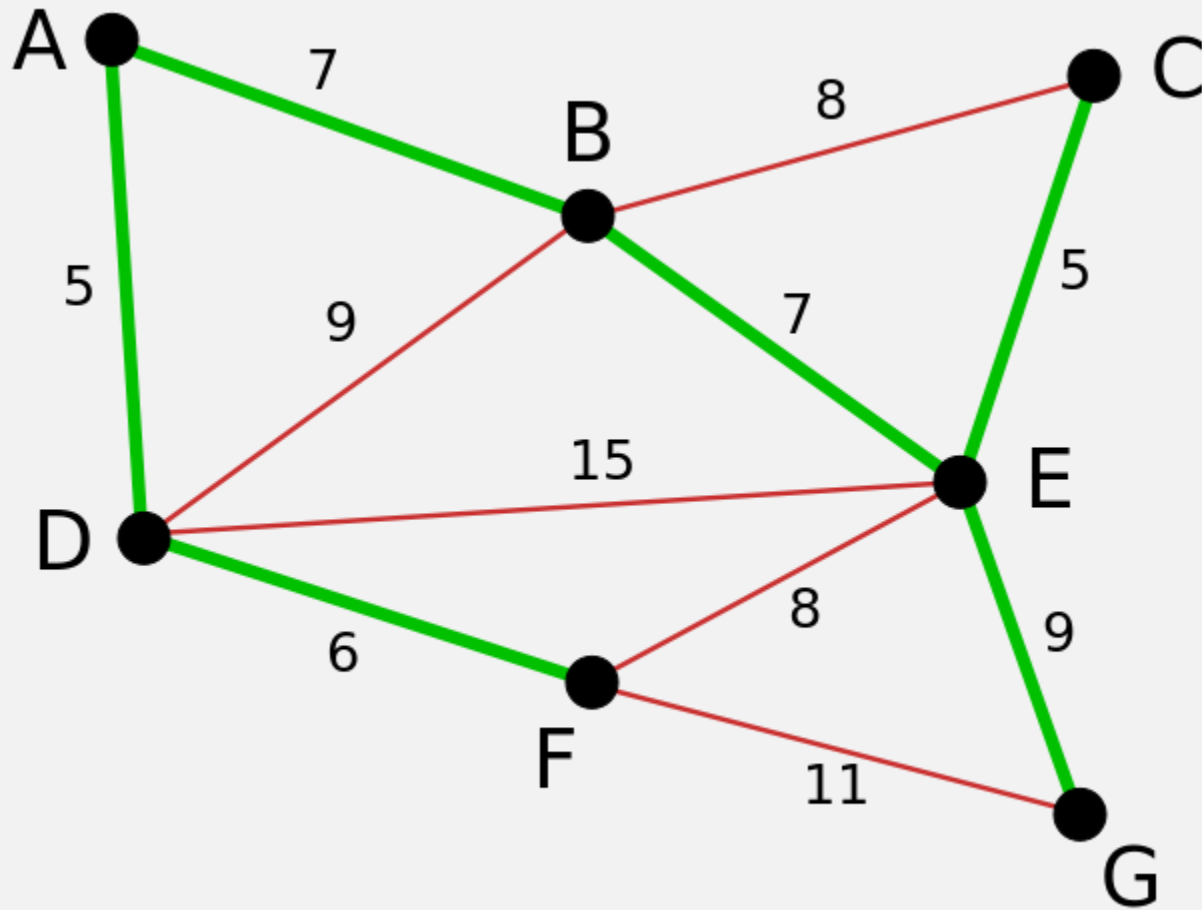
Kruskal algorithm



Kruskal algorithm



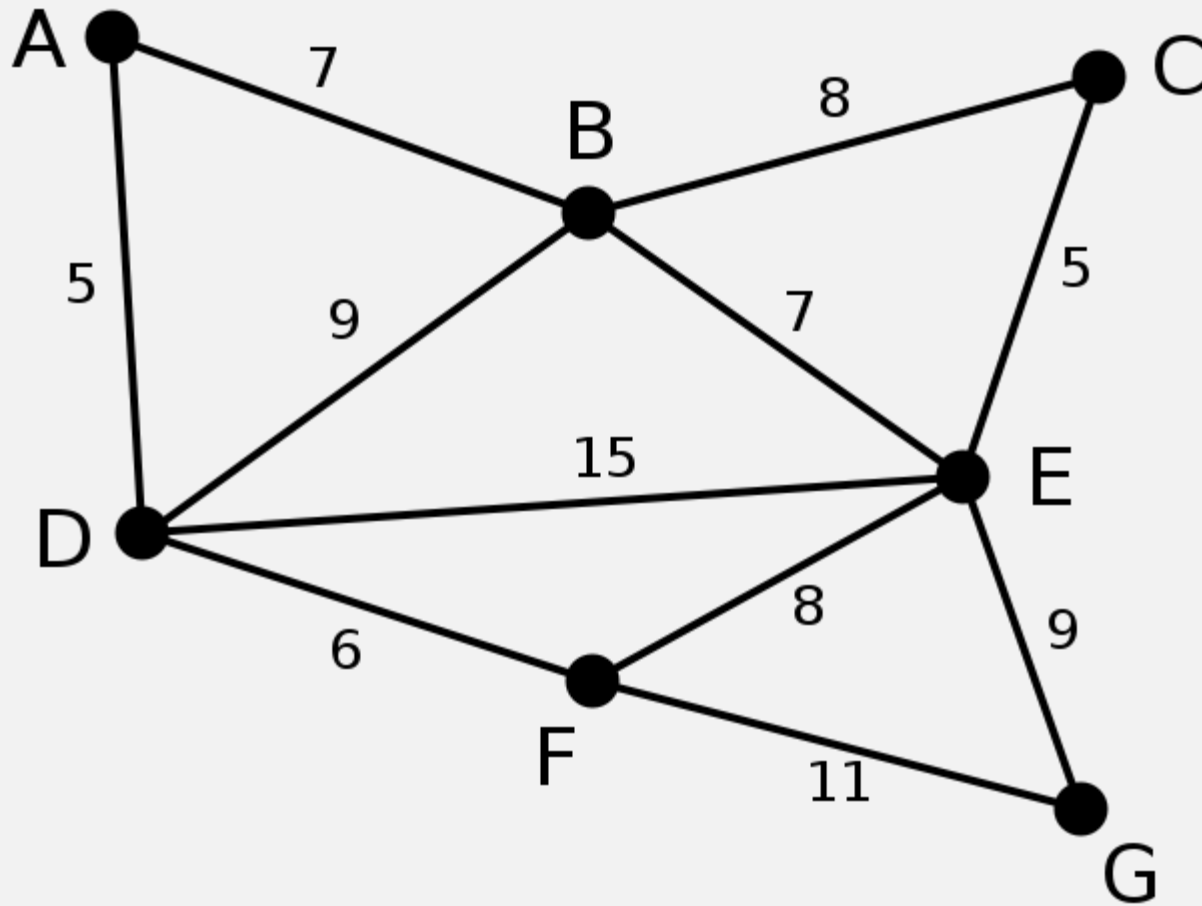
Kruskal algorithm



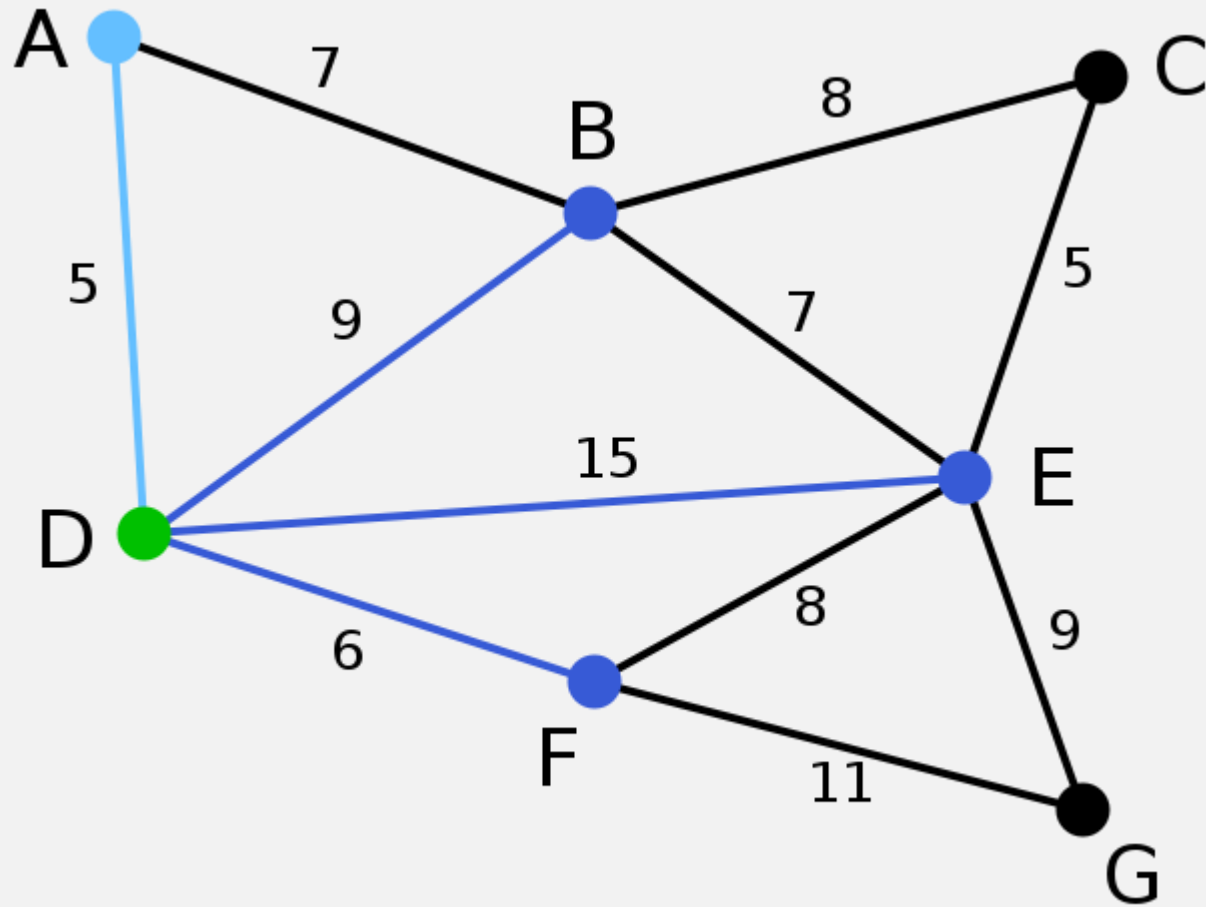
Prim algorithm

- ① 시작점 선택
- ② 트리에 연결된 간선 중 트리 내의
두 정점을 포함하지 않는 가장
작은 가중치의 간선 선택
- ③ 트리에 추가
- ④ 모든 정점이 트리에 포함되면 종료

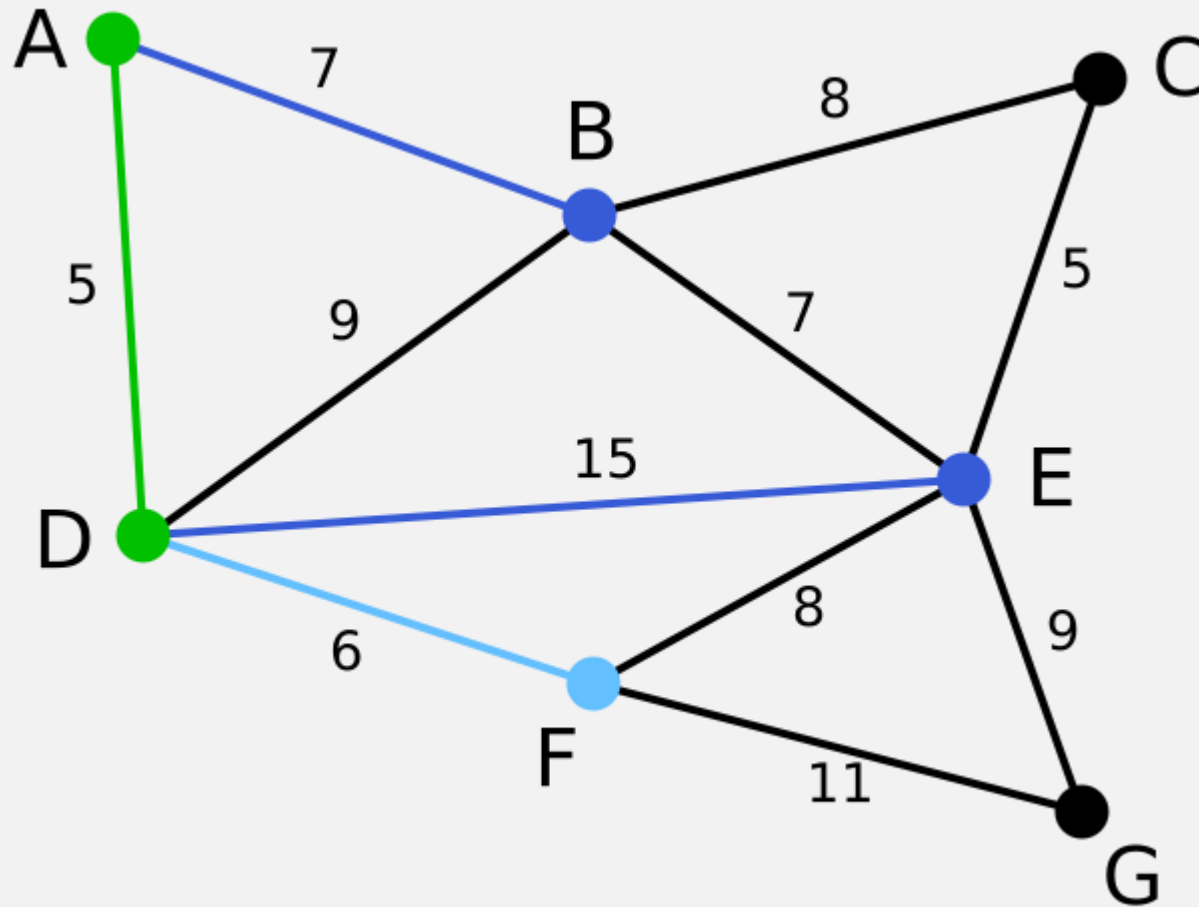
Prim algorithm



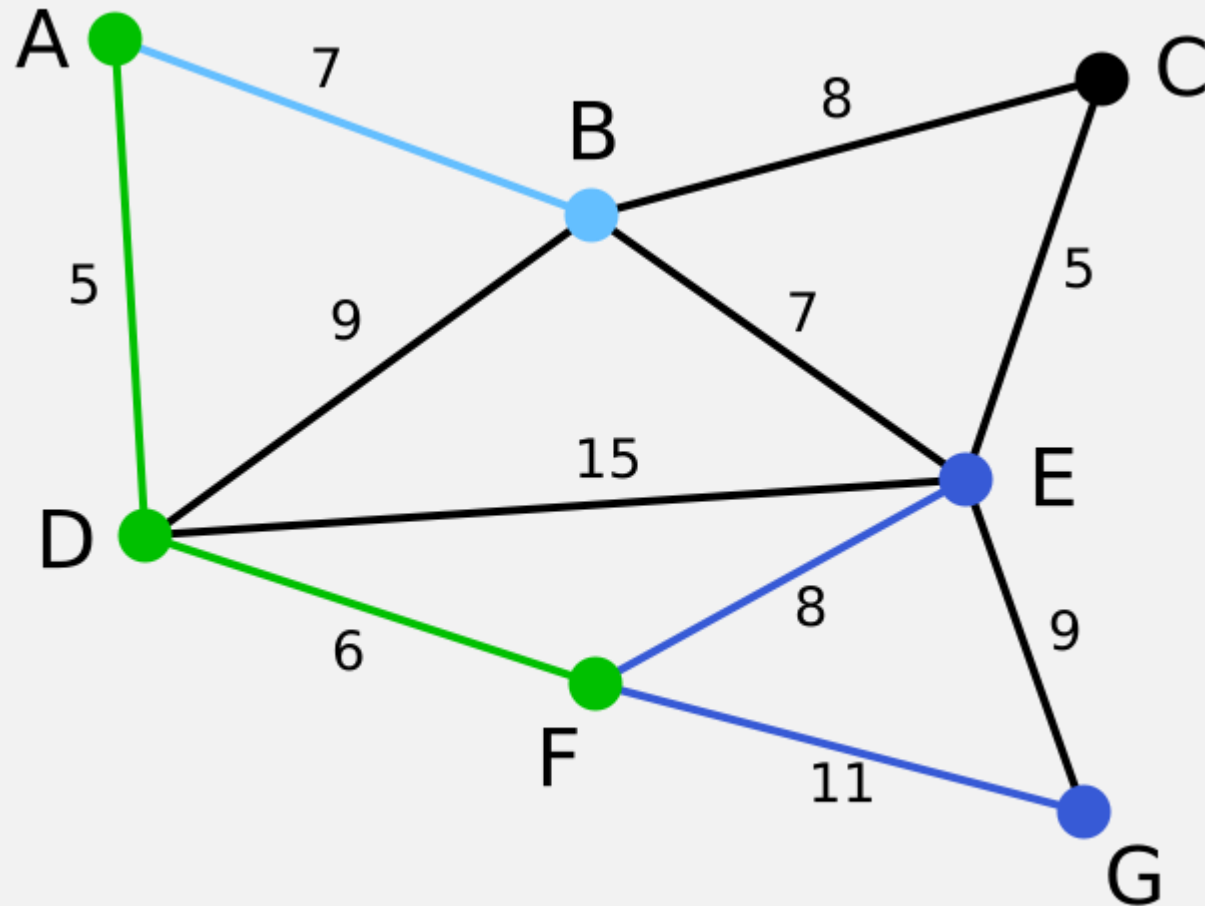
Prim algorithm



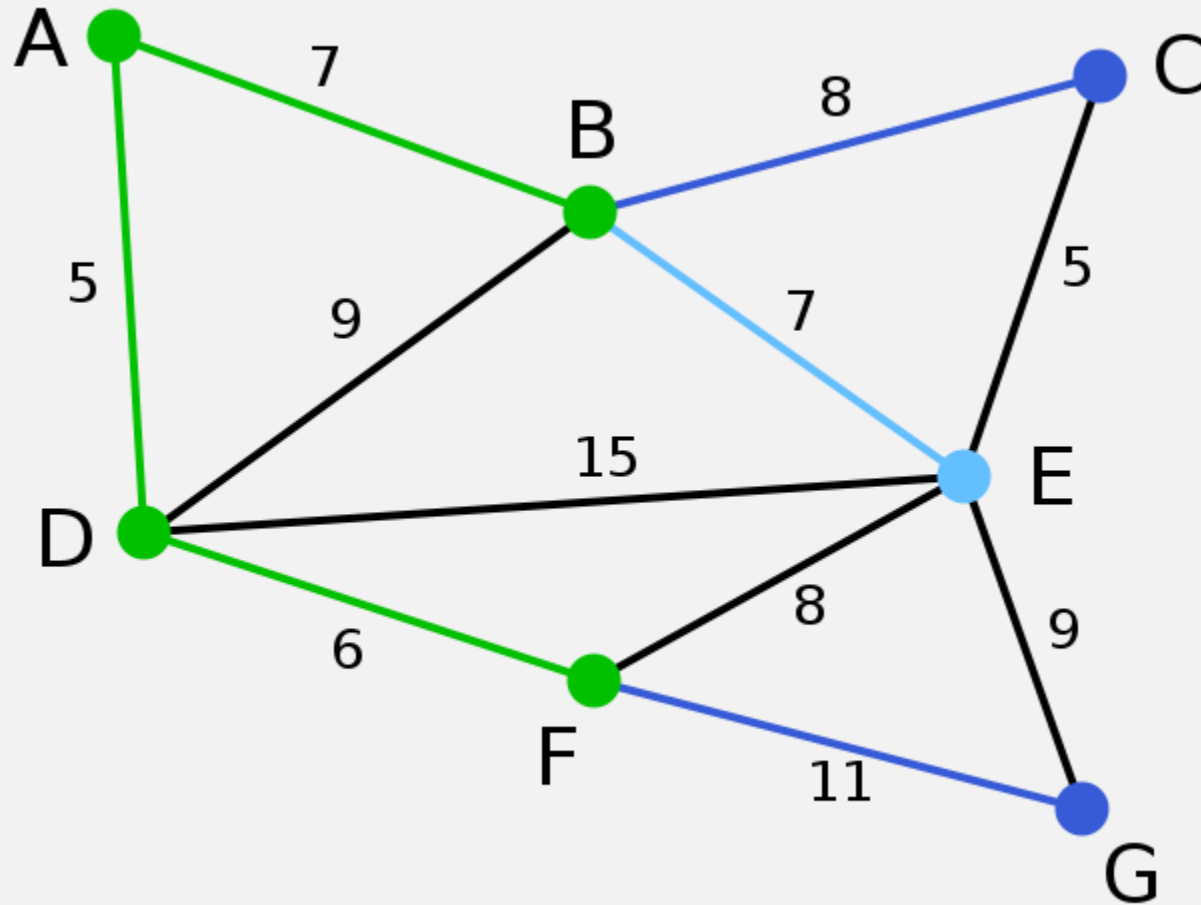
Prim algorithm



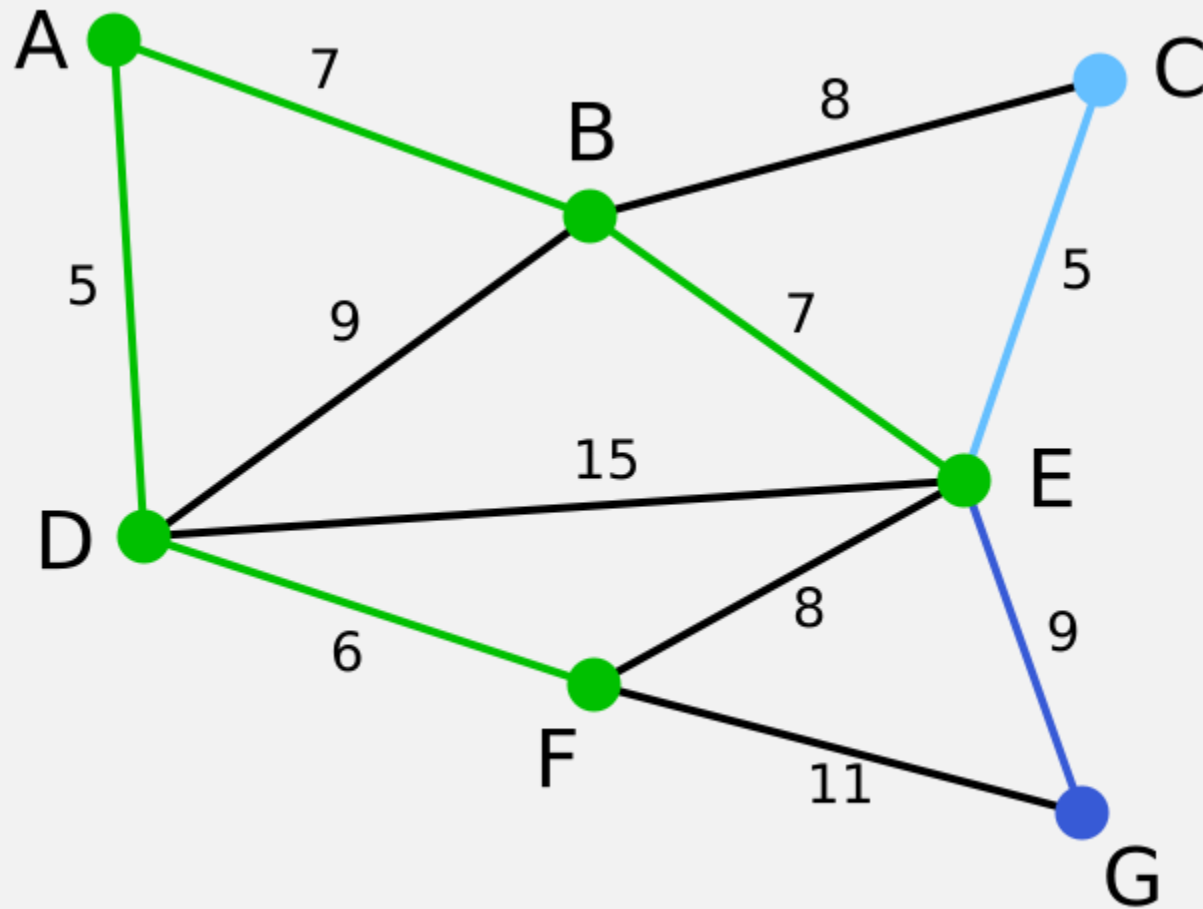
Prim algorithm



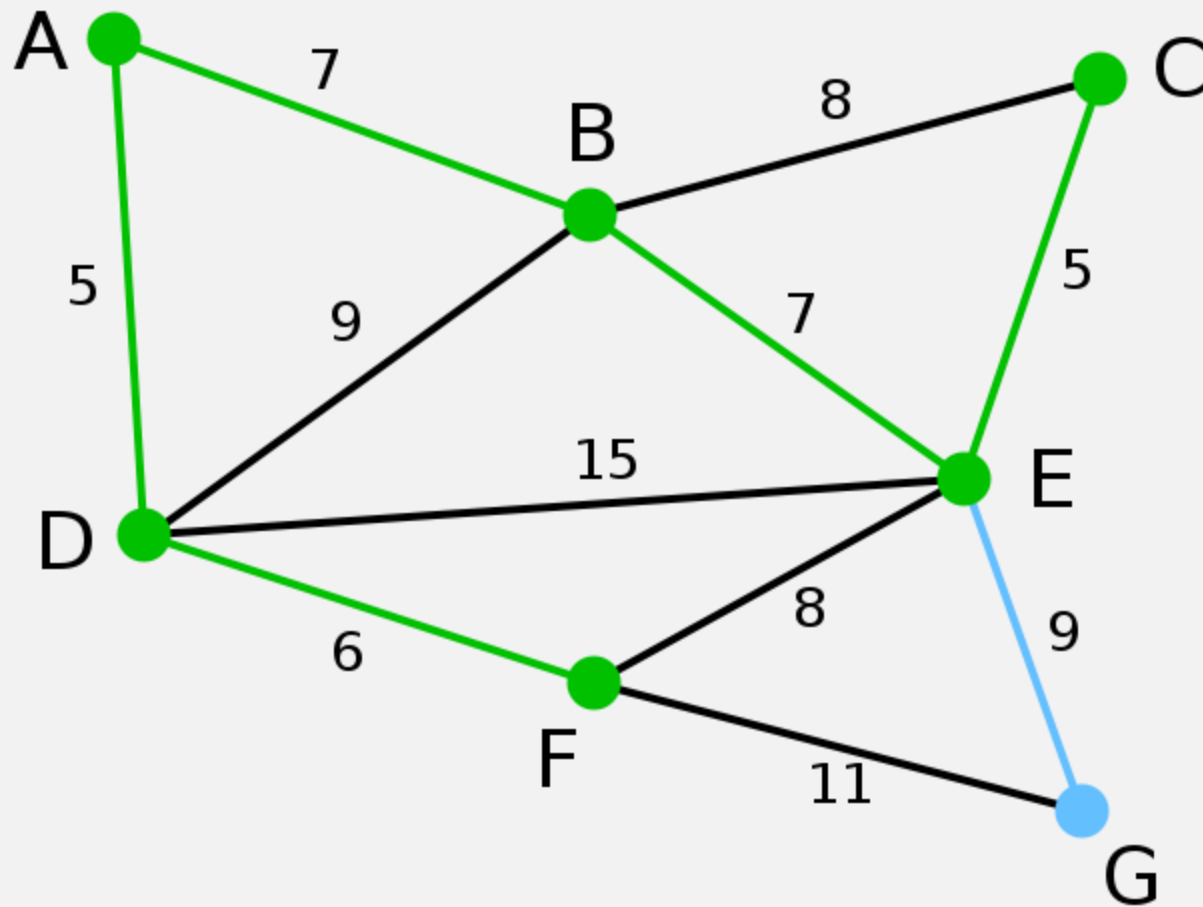
Prim algorithm



Prim algorithm



Prim algorithm



MST 관련문제

최소 스패닝 트리

Lan

Conscription

그 외 그래프 문제

트리의 지름

화이트칼라

Reference

- ✓ 2012 ACM-ICPC study 강의자료
- ✓ icpc.sogang.ac.kr/wiki
- ✓ JM book



END