Minimum Spanning Tree

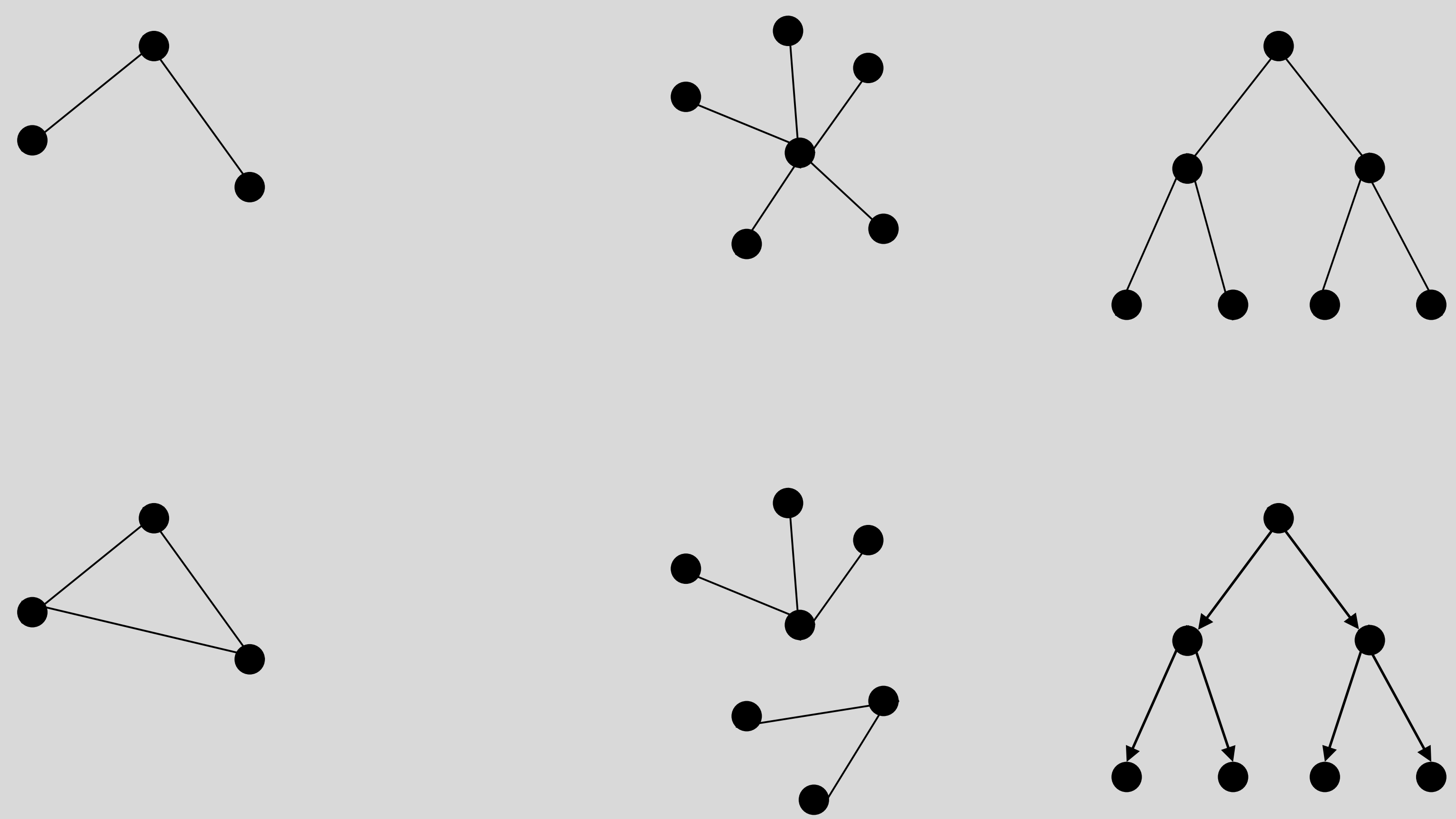
서강대학교 엄태경

Tree

- Acyclic, connected, undirected graph

Tree

Not Tree



Properties of Tree

Edge

- $|E| = |V| - 1$
- A tree with *n* vertices has *n-1* edges

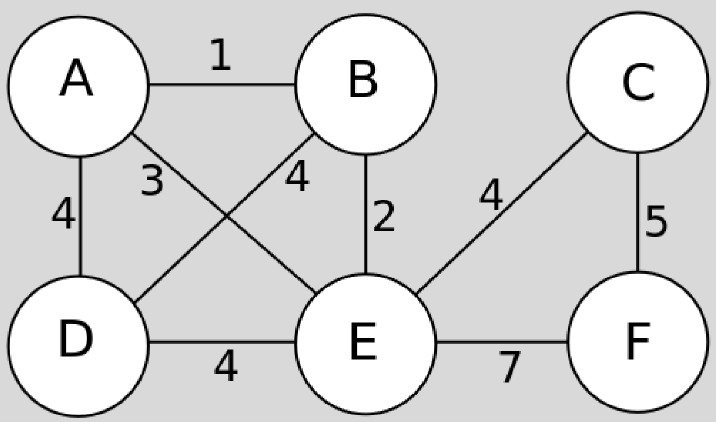
Path

- Any pair of two vertices are connected by exactly one path.
  - **Connected Graph**
    At least one path between two vertices

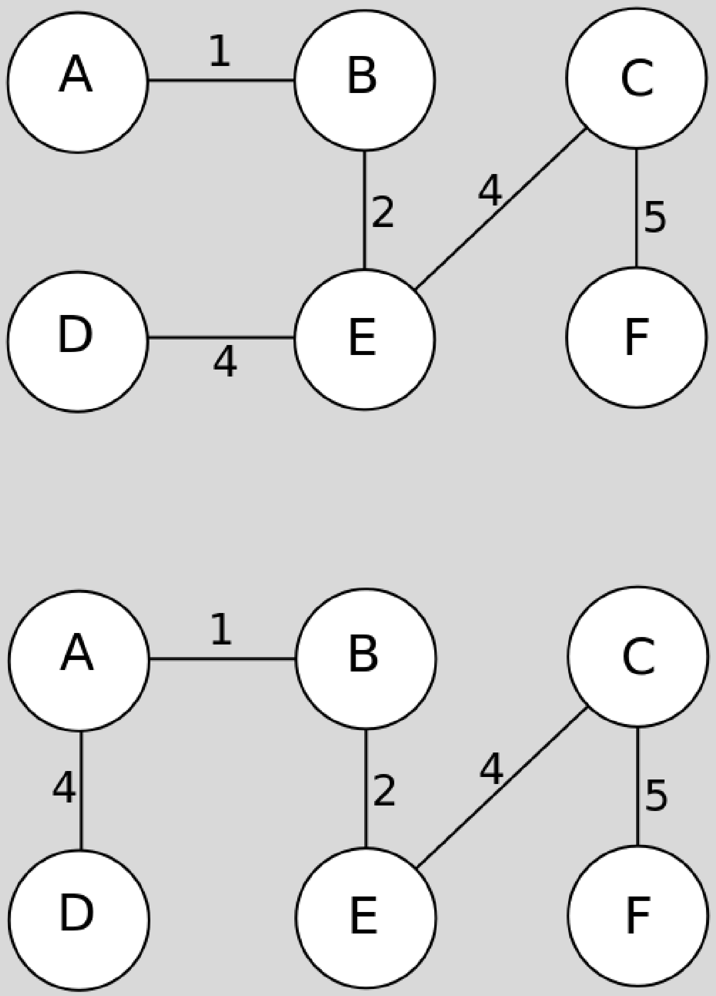
Spanning Tree of a Graph

Spanning Tree

- Any tree with same set of vertices as the graph



A graph

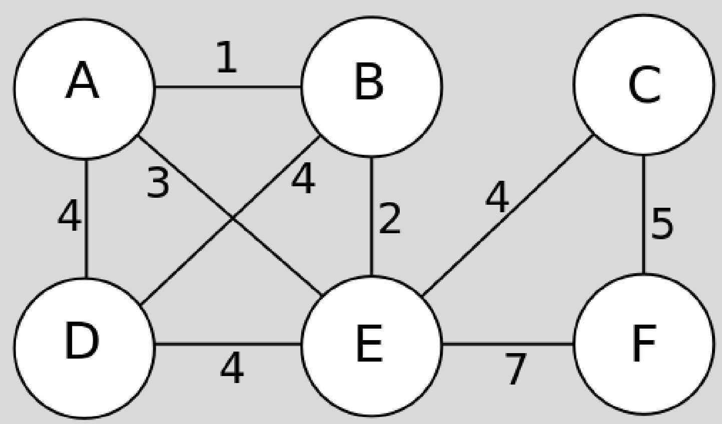


Two spanning trees

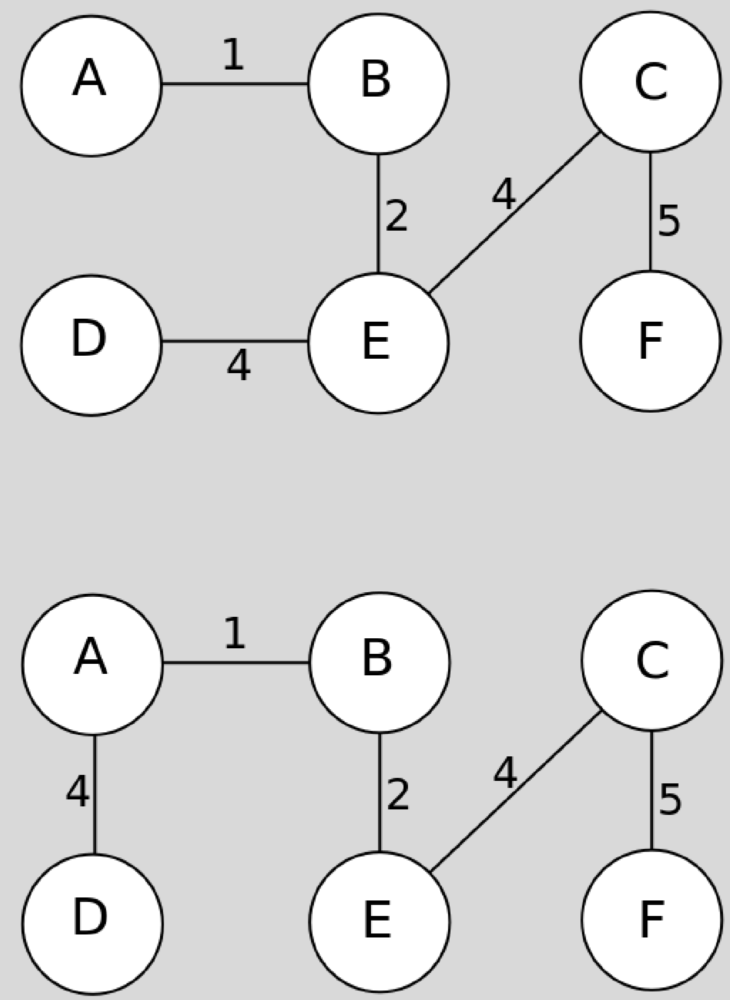
Minimum Spanning Tree

Minimum Spanning Tree

- Spanning tree with minimum weight



A graph



Minimum spanning trees

1922 네트워크 연결

- 컴퓨터 N ($1 \le N \le 1000$) 대를 모두 연결하고 싶다.
- 선 하나로 컴퓨터 두 대를 연결할 수 있다.
- 컴퓨터 a와 컴퓨터 b가 연결이 되어 있다는 말은 a에서 b로의 경로가 존재한다는 것을 의미한다. A에서 b를 연결하는 선이 있고, b와 c를 연결하는 선이 있으면 a와 c는 연결이 되어 있다.
- 두 컴퓨터를 연결할 수 있는 경우의 수는 M ($1 \le M \le 100{,}000$)
- 각 연결마다 비용이 책정 되어있다. ($1 \le c \le 10{,}000$)
- 모든 컴퓨터를 연결할 수 있는 비용의 합을 최소값은?

Naïve Algorithm

Brute Force?

1. For every possible spanning tree:
   2. Calculate the sum of weights
   3. Keep the minimum sum

Time Complexity

- 1: Complete graph has $|V|^{|V|-2}$ spanning trees
- 2: A spanning tree has $|V|-1$ edges

➔ $O(|V|-1 \times V^{V-2}) = O(|V|^{|V|})$

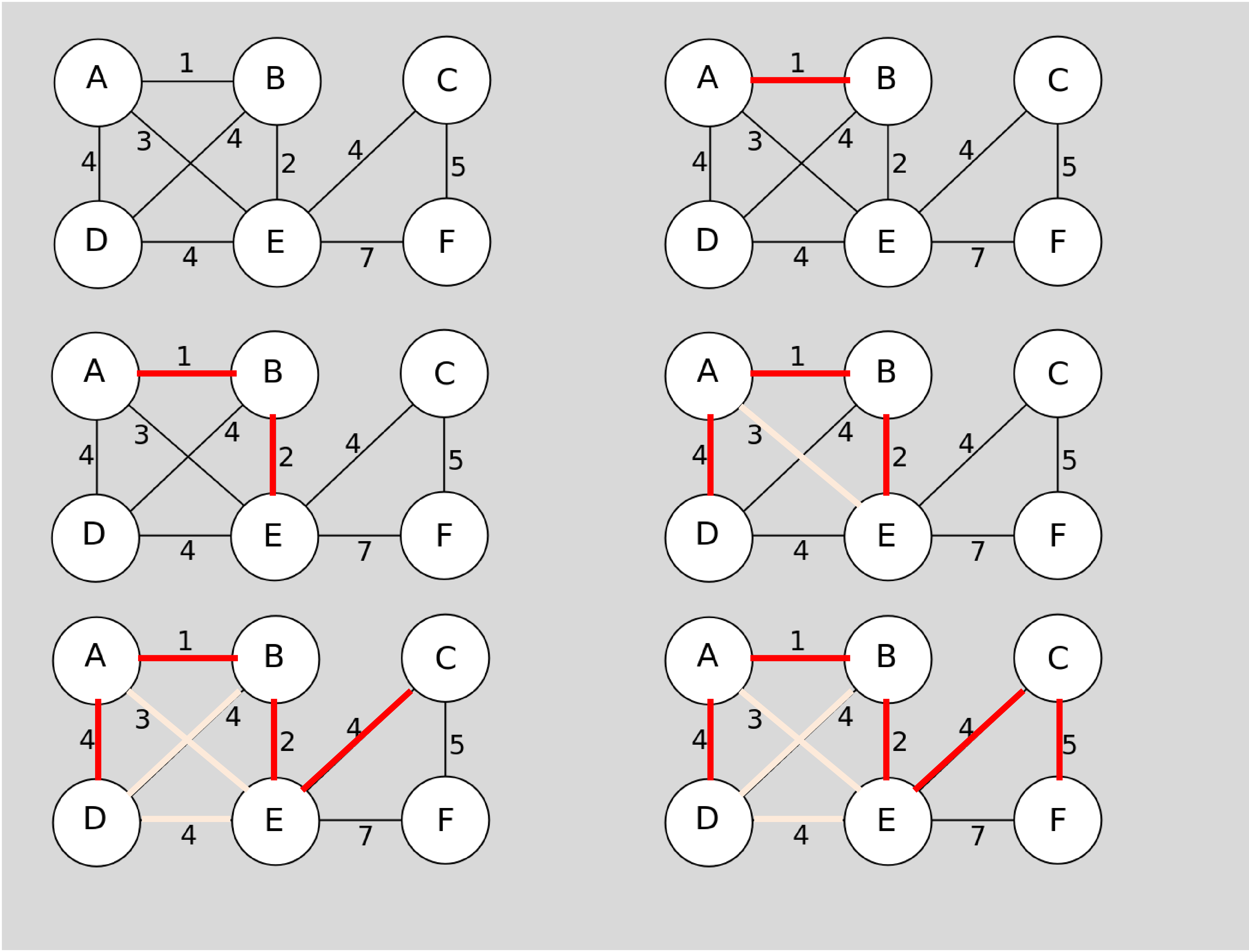
Kruskal’s Algorithm (1)

Greedy Algorithm

- Continue selecting edges in increasing order of weights *if* an edge does not make a cycle.
- Repeat until |V|-1 edges are selected.

Kruskal's Algorithm (2)

Example



Kruskal's Algorithm (3)

Kruskal's

```
1 A = ∅
2 for each v in V
3    MAKE-SET(v)
4 Sort E by increasing order of weights
5 for each edge (u, v) in E:
6    if FIND-SET(u) != FIND-SET(v) :
7       A = A ∪ {(u, v)}
8       UNION(u, v)
9 return A
```

Time Complexity

1: $O(1)$ 2: $|V|$ iter. 3: $O(1)$
4: $O(|E| \log|E|)$ 5. $|E|$ iterations
6: $O(1)$ in practice 7: $O(1)$
8: $O(1)$ in practice

$\rightarrow O(|E| \log |E|) = O(|E| \log |V|)$

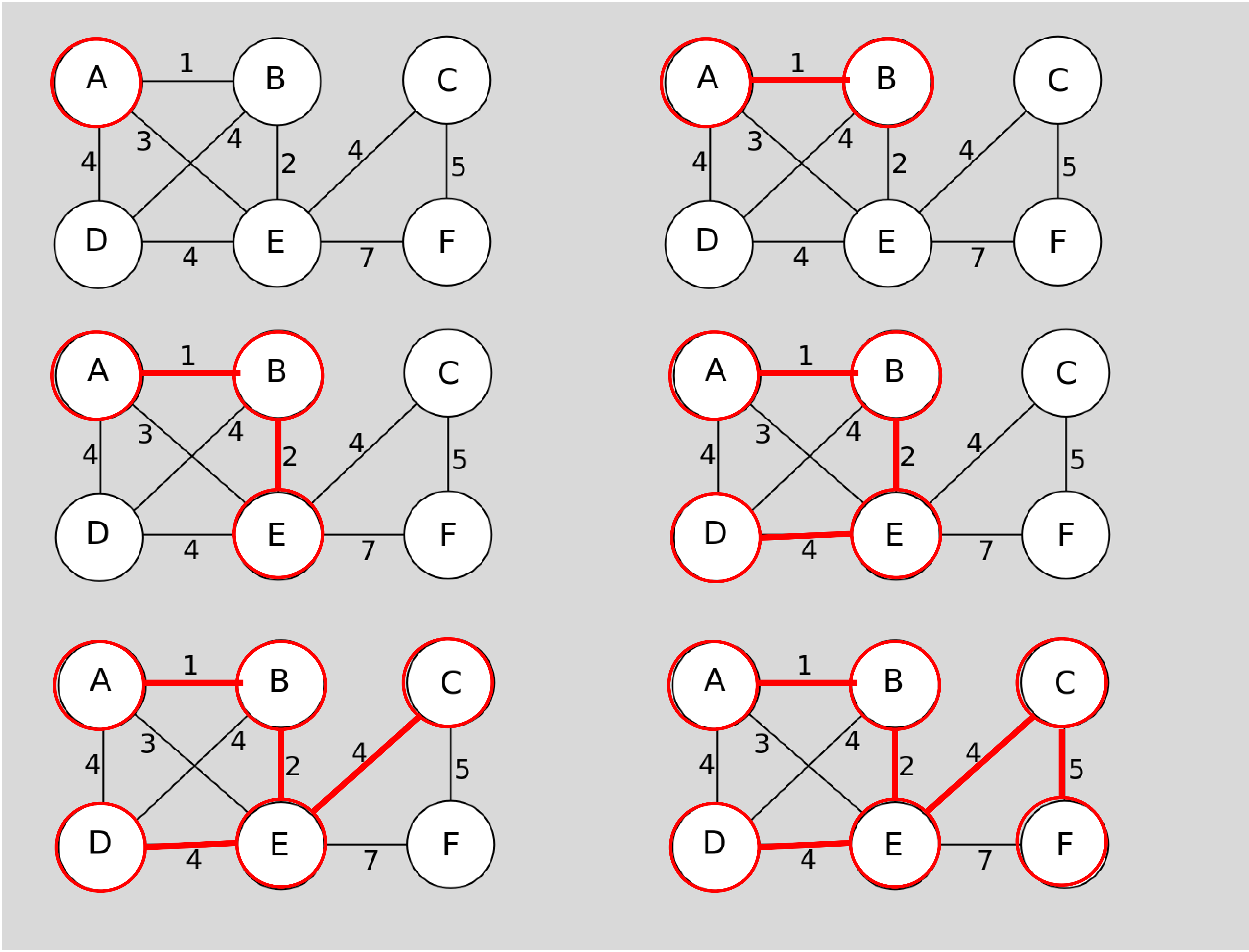
Prim's Algorithm (1)

Greedy Algorithm

- Add a random starting vertex to tree T
- Find the shortest edge (u, v) that connects T and G-T
- Add the edge to T
- Continue until T has |V| vertices

Prim's Algorithm (2)

Example



Prim’s Algorithm (3)

Prim’s

```
define f[v]: true if and only if vertex v is added to MST.
define parent[v]: ‘parent’ vertex of v.
define key[v]: cost of minimum known edge to v.
1 for each v in V
2    f[v] = false
3 f[v0] = true, where v0 is the starting vertex
4 key[v0] = 0.
5 Initialize a priority queue pq of pair (weight, vertex)
6 Add v0 to pq
7 while pq is not empty:
8     u := minimum vertex in pq .
9     Pop pq
10    f[u] = true
11    for each neighbor w of u:
12       if !f[w]:
13           parent[w] = u
14           key[w] = cost(u,w)
15           Insert (key[w], w) into pq
```

Time Complexity

Line 15 is called at most |E| times. pq insertion is O(log|E|)

→ O(|E| log|E|)

Practice

BOJ

- 1922 네트워크 연결
- 1197 최소 스패닝 트리
- 6497 전력난
- 1647 도시 분할 계획
- 4343 Arctic Network
- 9373 복도 뚫기