

Spanning Tree

= 그래프의 모든 정점을 연결하는 트리

최소 신장 트리

= 여러 간선 중, 가중치 합이 최소가 되는 간선만 남은 트리

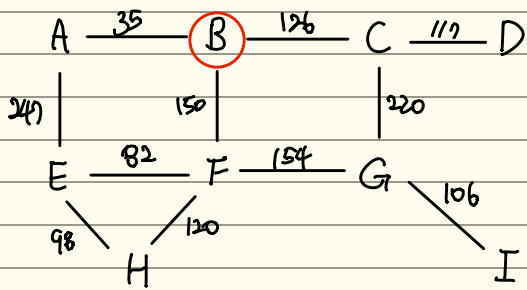
최소 신장 트리는 연결된 가중치 그래프로부터 만들어진다

최소 신장 트리 생성 알고리즘

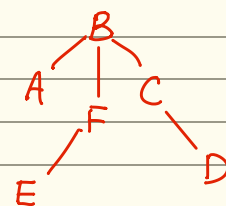
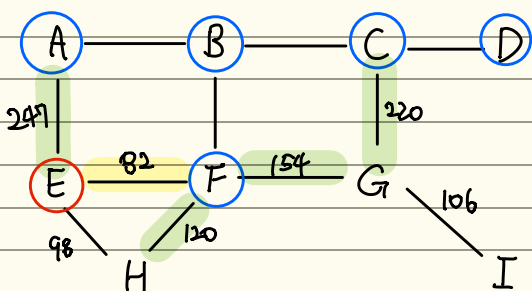
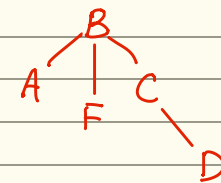
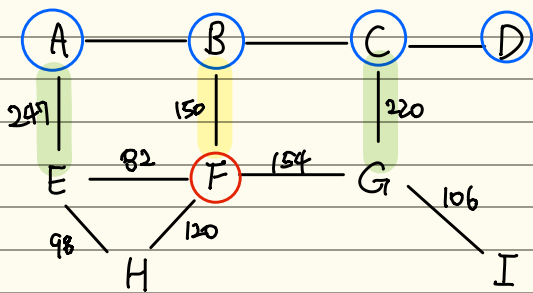
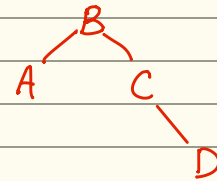
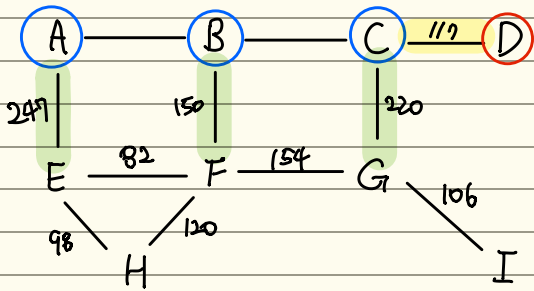
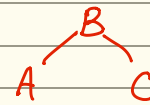
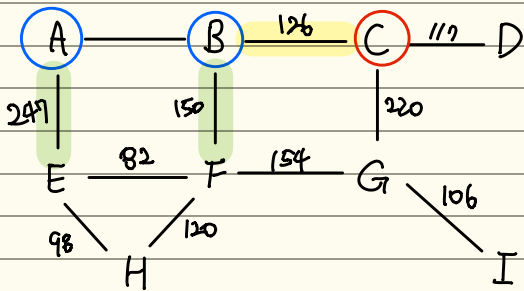
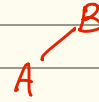
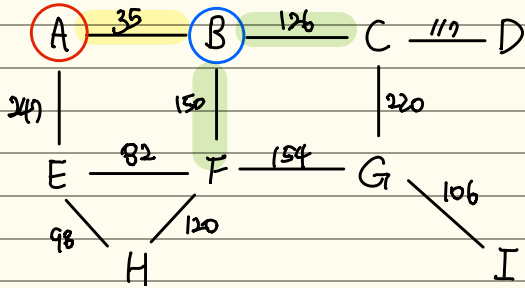
① Prim algorithm

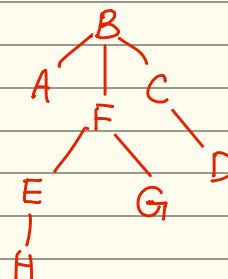
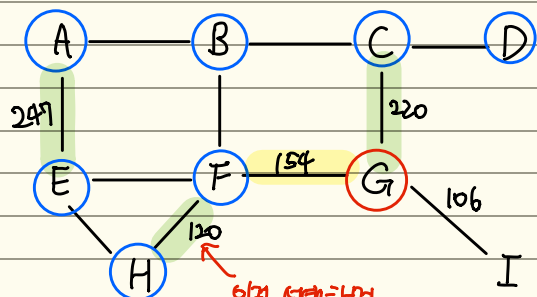
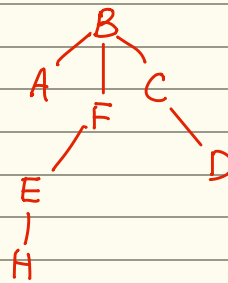
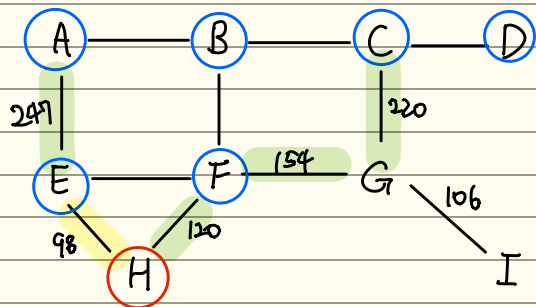
- 1) 그래프에서 임의의 정점 선택 → 최소 신장 트리의 Root Node
- 2) 최소 신장 트리에 삽입된 정점에서 이어진 모든 가중치 조사
→ 가장 작은 가중치 간선 선택하여 이어진 정점을 최소 신장 트리에 삽입
단) 새로 넣는 노드는 기존 최소 신장 트리의 노드와 사이클을 형성해서는 안된다
- 3) 2)를 반복하고 모든 노드를 방문했다면 종료

임의 노드 선택

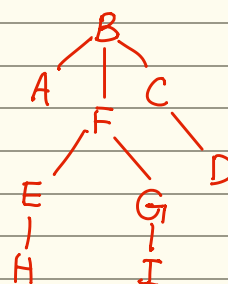
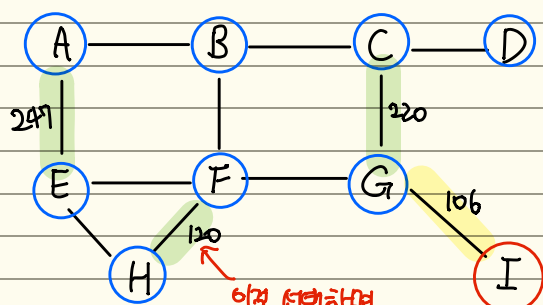


B





이걸 선택하면
사이클이 생김
(isVisited == True) 인 경우 제외



이걸 선택하면
사이클이 생김
(isVisited == True) 인 경우 제외

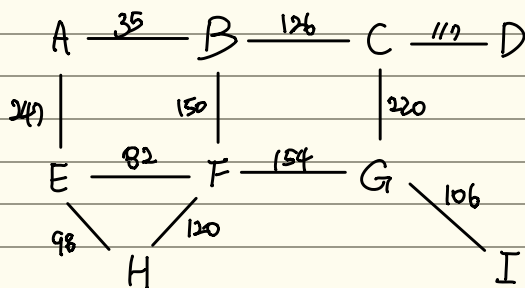
② Kruskal's Algorithm

- 1) 그래프 내 모든 간선을 가중치에 따라 오름차순 정렬하여 List 생성
- 2) 1)에서 만든 리스트를 순회하면서 간선을 최소인장트리에 추가
이때 추가한 간선이 사이클을 형성하는 안됨

어떻게 사이클을 형성하는지 판단하나?

⇒ Union-Find 또는 서로소 집합 자료구조를 이용한다.

- 1) 정점별로 각각의 분리집합을 만들고, 간선으로 연결된 정점들에 대해서는 합집합으로 만든다
- 2) 간선으로 연결할 두 정점이 같은 집합 ⇒ Cycle 발생



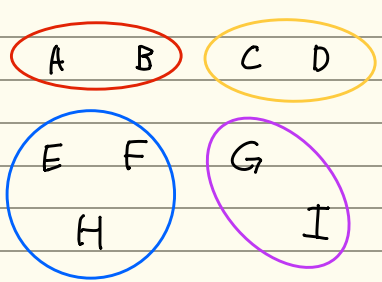
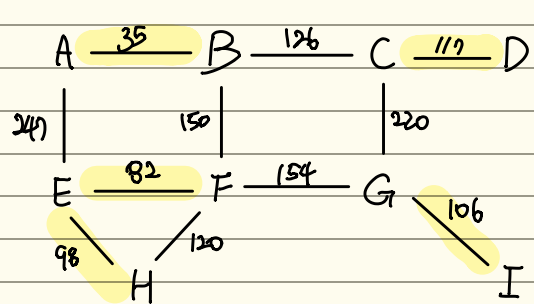
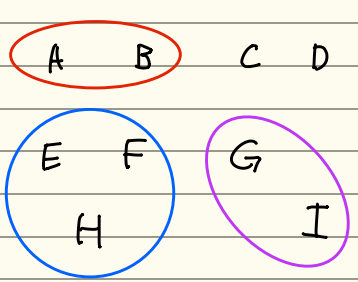
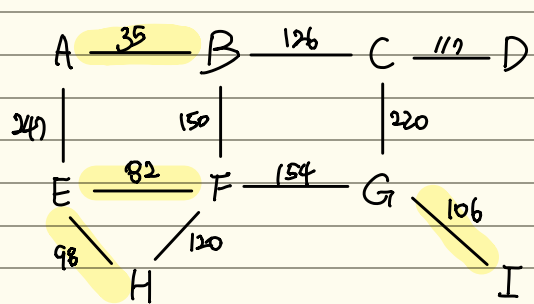
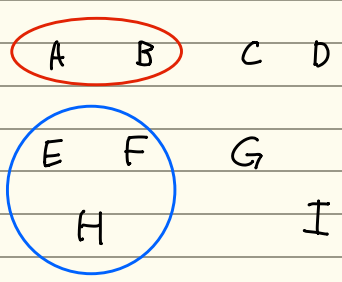
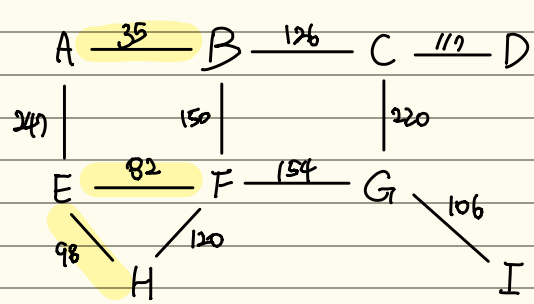
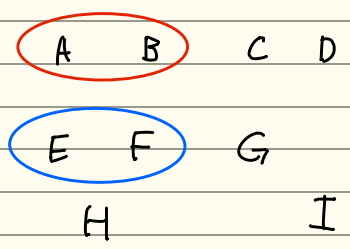
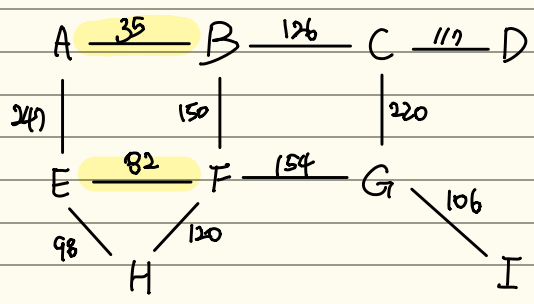
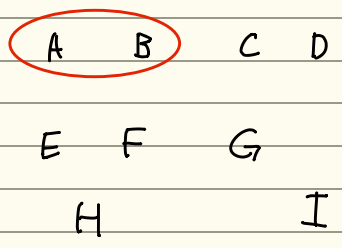
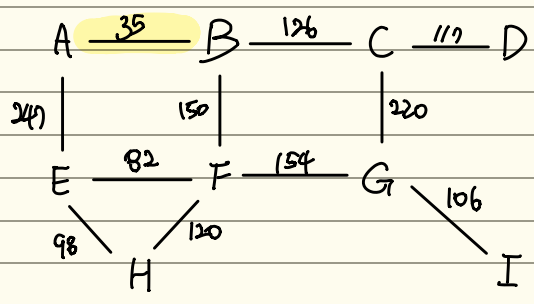
⇒

A-B : 35
E-F : 82
E-H : 98
G-I : 106
C-D : 117
F-H : 120
B-C : 126
B-F : 150
F-G : 154
C-F : 162
C-G : 220
A-E : 247

⇒

A	B	C	D
E	F	G	
	H		I

example



FH는 {EFH}에 속하므로 Cycle 발생 \Rightarrow pass

