

# Kruskal's algorithm.

→ Cycle property 이용

: 만약 Cycle의 최대 비용 간선은 MST가 포함되지 않는다.

0.  $MST = \{ \}$

1. 간선은 비용의 오름 차순으로 정렬.  $O(m \log m)$

2. 각각의 간선에 대해서 양쪽 노드  $e = (u, v)$ :

if  $MST \cup \{e\}$  has no cycle:  $\leftarrow C$

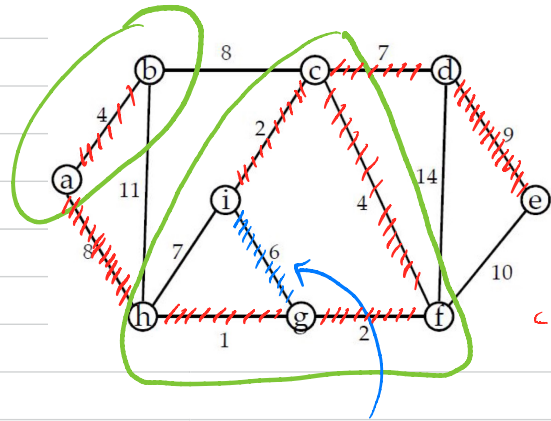
↳  $MST = MST \cup \{e\}$

if  $|MST| == n-1$ :

↳ break

return MST

$MST = \text{Forest} = \text{트리들의 집합.}$



$m$ 번  
=  $O(m)$

⇒  $O(m \log m) + O(m)$

Cycle을 방지 하기 위해서 → Union-find 구조 사용.

1) 내가 속한 집합이 무엇인지 알려 주고, : find

2) 서로 다른 집합을 Union 해서 합집합을 만들고

6을 넣을 순간 사이클 발생

Cycle property: 2개의 노드 사이에는 어떤 사이클도

생성하지 않고, 그 사이클이 있는

최대 비용 간선은 MST에 포함되지

않는다. 6을 선택했을 때 사이클이

생기기 때문에 6이 가장 최근에 포함 되듯이

때문에 MST를 만족 X

총 4번의 find + 1번의 Union = 5회  
 $O(m \log n)$

KRUSKAL( $G = (V, E)$ ):

1  $F = \emptyset$

2 for each  $v \in V$ :

3 MAKE-SET( $v$ )  $O(1)$

4 for each  $e=(u,v)$  ordered by weight( $u,v$ ) increasing:

5 if FIND-SET( $u$ )  $\neq$  FIND-SET( $v$ ): →  $u$ 와  $v$ 가 같은 집합이냐? = 같은 SubTree에 존재 하냐?

6  $F = F + e$

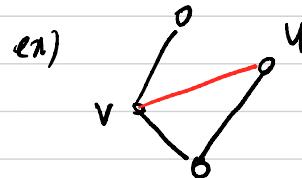
7 UNION(FIND-SET( $u$ ), FIND-SET( $v$ )) #union-find

8 return F if  $|F| == n-1$ : break

간선에 연결된  $u, v$  노드

→  $O(m \log m)$

→  $u$ 와  $v$ 가 속한 집합이 같으면 새로 집  
놓으면 사이클이 생긴다



- Initial State:  $\{a\}, \{b\}, \{c\}, \{d\}, \{e\},$

$\{f\}, \{g\}, \{h\}, \{i\}$

$\{a,b\}, \{c,d\}, \{e,f\}, \{g,h\}, \{i\}$

$\{a,b,c\}, \{d,e,f\}, \{g,h,i\}$

$\{a,b,c,d,e,f\}, \{g,h,i\}$

$\{a,b,c,d,e,f,g,h,i\}$

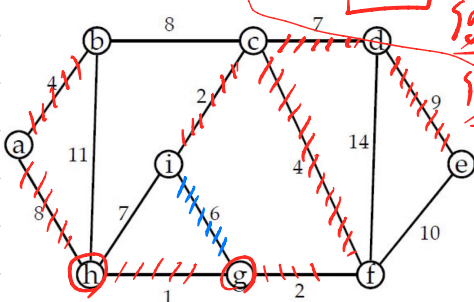
$\{a,b,c,d,e,f,g,h,i\}$

$\{a,b,c,d,e,f,g,h,i\}$

$\{a,b,c,d,e,f,g,h,i\}$

$\{a,b,c,d,e,f,g,h,i\}$

$\{a,b,c,d,e,f,g,h,i\}$



$= \{a,b,c,d,e,f,g,h,i\}$

총 4개 정점

총:  $O(m \log m + m \log n)$   $m \times m \log n$

$= O(m \log m + m \log m)$

$= O(m \log m)$