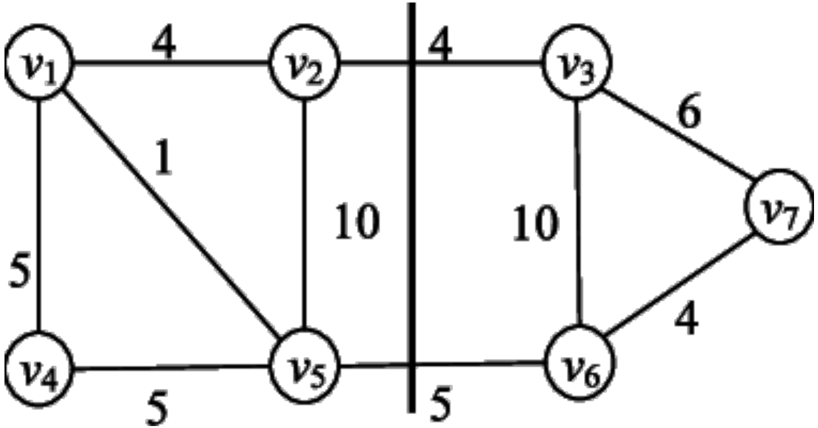
MINIMUM SPANNING TREES – PRIM'S ALGORITHM

CS340

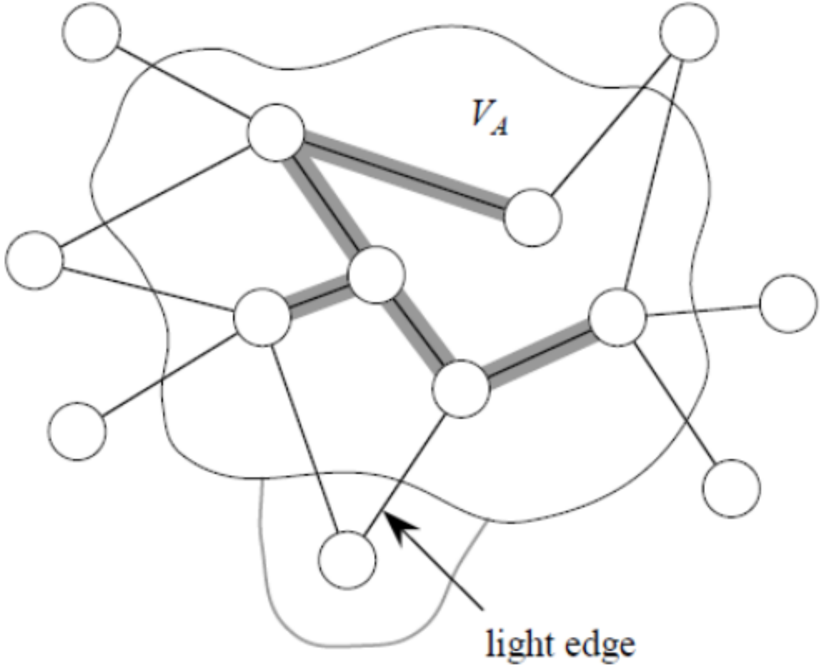
Crossing a cut proof



The light edge crossing any cut is part of the MST.

Prim's Algorithm

- Builds one tree, so A is always a tree.
- Starts from an arbitrary "root" r.
- At each step, find a light edge crossing cut (V_A, V-V_A), where V_A = vertices that A is incident on. Add this edge to A.



Prim's Algorithm

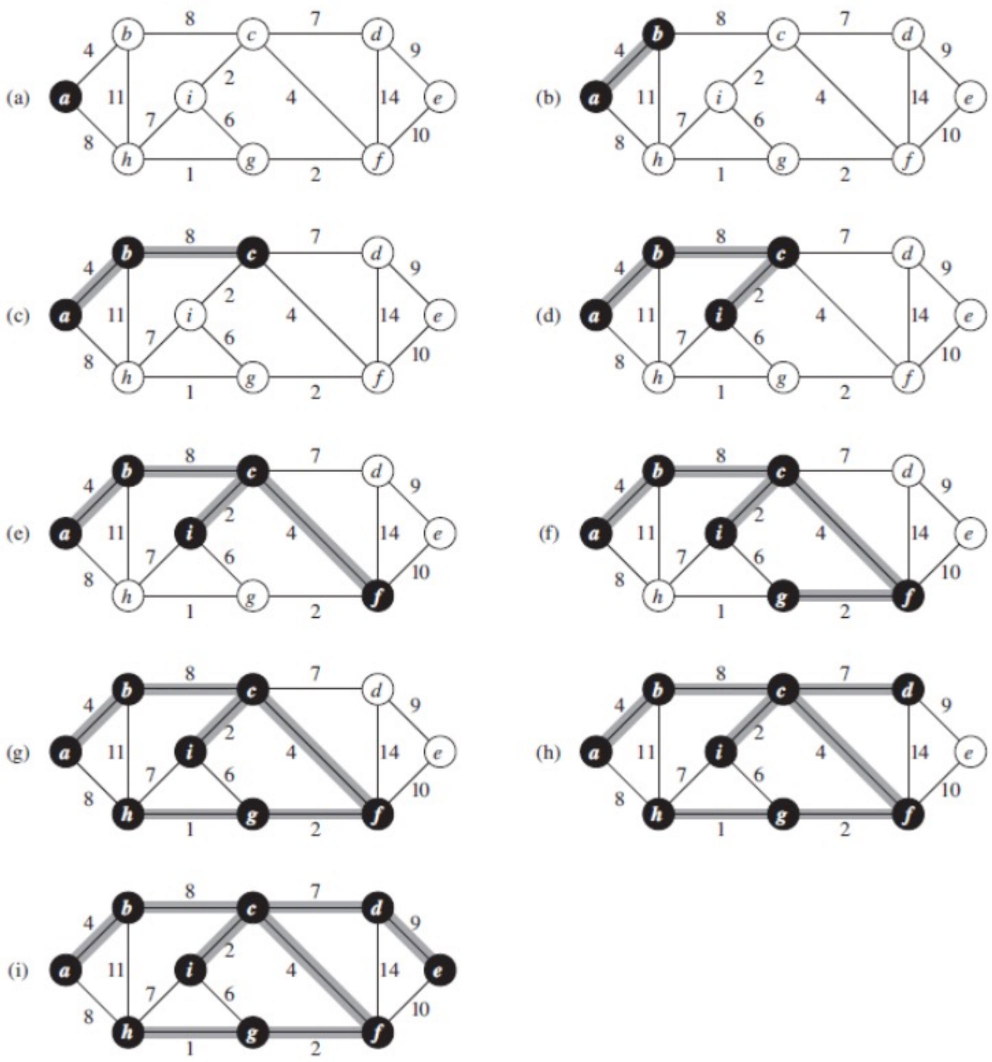
- How to find the lightest edge quickly?
  Use a priority queue:
  - Each object is a vertex in $V - V_A$.
  - Key is minimum weight of any edge (u,v) where $u \in V_A$.
  - Then the vertex returned by EXTRACT-MIN is v such that there exists $u \in V_A$ and (u,v) is light edge crossing $(V_A, V-V_A)$.
- Key of v is ∞ if v is not adjacent to any vertices in V_A.

Prim's Algorithm

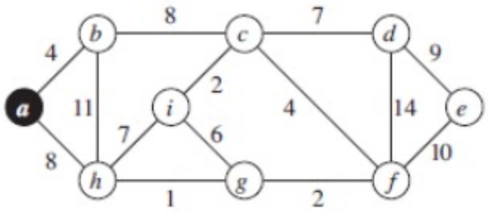
PRIM(G, w, r)

```
Q = ∅
for each u ∈ G.V
    u.key = ∞
    u.π = NIL
    INSERT(Q, u)
DECREASE-KEY(Q, r, 0)        // r.key = 0
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    for each v ∈ G.Adj[u]
        if v ∈ Q and w(u, v) < v.key
            v.π = u
            DECREASE-KEY(Q, v, w(u, v))
```

Prim's Algorithm



Prim's Algorithm

(a) 

parent									
node	a	b	c	d	e	f	g	h	i
init									
step1									
2									
3									
4									
5									
6									
7									

Prim Efficiency

- Suppose Q is a priority queue.
- Initialize Q and first for loop: O(V lg V)
- Decrease key of r: O(lg V)
- while loop:
- |V| EXTRACT-MIN calls = O(V lg V) ≥ |E| DECREASE-KEY calls = O(E lg V)
- Total: O(E lg V)

PRIM(G, w, r)
  $Q = \emptyset$
  for each $u \in G.V$
    $u.key = \infty$
    $u.\pi = \text{NIL}$
    INSERT(Q, u)
  DECREASE-KEY$(Q, r, 0)$ // $r.key = 0$
  while $Q \neq \emptyset$
    $u =$ EXTRACT-MIN(Q)
    for each $v \in G.Adj[u]$
      if $v \in Q$ and $w(u, v) < v.key$
        $v.\pi = u$
        DECREASE-KEY$(Q, v, w(u, v))$