CS473-Algorithms I

Lecture ?

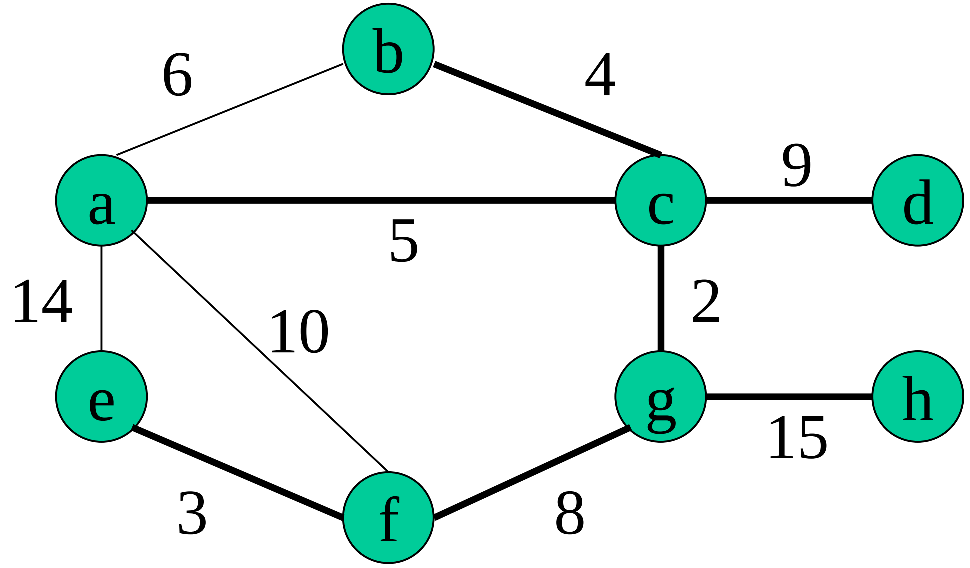
Minimum Spanning Tree

Minimum Spanning Tree

- One of the most famous greedy algorithms.
- Undirected Graph $G = (V, E)$
- Connected
- Weight Function $\omega : E \rightarrow R$
- Spanning Tree : Tree that connects all vertices

Minimum Spanning Tree

- MST : $\omega(T) = \Sigma_{(u,v) \in T}\ \omega(u,v)$
- Note : MST is not unique.
- Sample :



Optimal Structure

- Optimal Structure : Optimal tree has optimal subtrees.

  - Let T be an MST of $G = (V, E)$
  - Removing any edge (u,v) of T partitions T into two subtrees : T_1 & T_2

  Where $T_1 = (V_1, E_{T_1})$ & $T_2 = (V_2, E_{T_2})$

Optimal Structure

- Let $G_1 = (V_1,E_1)$ & $G_2 = (V_2,E_2)$ be subgraphs induced by V_1 & V_2

  i.e. $E_i = \{ (x,y) \in E : x, y \in V_i$

- Claim : T_1 & T_2 are MSTs of G_1 & G_2 respectively

- Proof : $\omega(T) = \omega(u,v) + \omega(T_1) + \omega(T_2)$

- There can't be better trees than T_1&T_2 for G_1&G_2
- Otherwise, T would be suboptimal for G

Generic MST Algorithm

GENERIC-MST (G, ω)

A ← Ø

while A does not form a spanning tree **do**

find a safe edge (u,v) for A

A ← A ∪ { (u,v) }

return A

- A is always a subset of some MST(s)
- (u,v) is a safe edge for A if A ∪ { (u,v) } is also a subtree of some MST

Generic MST Algorithm

- One safe edge must exist at each step since :

- $A \subset T$ where T is an MST

- Let $(u,v) \in T \ni (u,v) \notin A \Rightarrow (u,v)$ is safe for A

- A cut (S, V - S) of G = (V , E) is a Partition of V
- An edge $(u,v) \in E$ crosses the cut (S, V - S)

  if $u \in S$ & $v \in V\text{-}S$ or vice versa
- A cut respects the set A of edges if no edge in A crosses the cut
- An edge is a light edge crossing a cut
  - If its weight is the minimum of any edges crossing the cut
  - There can be more than one light edge crossing the cut in the case of ties.

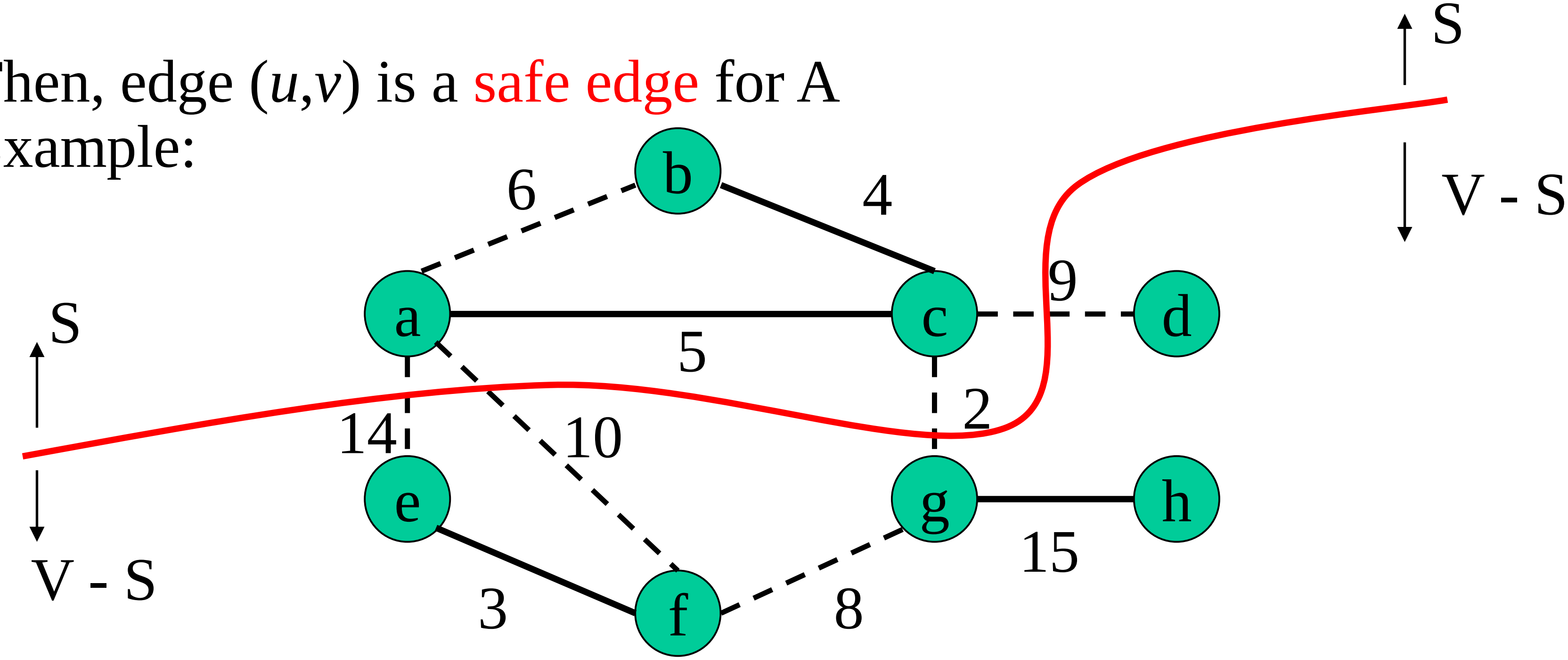
Properties of Generic MST Algorithm

THM1 :

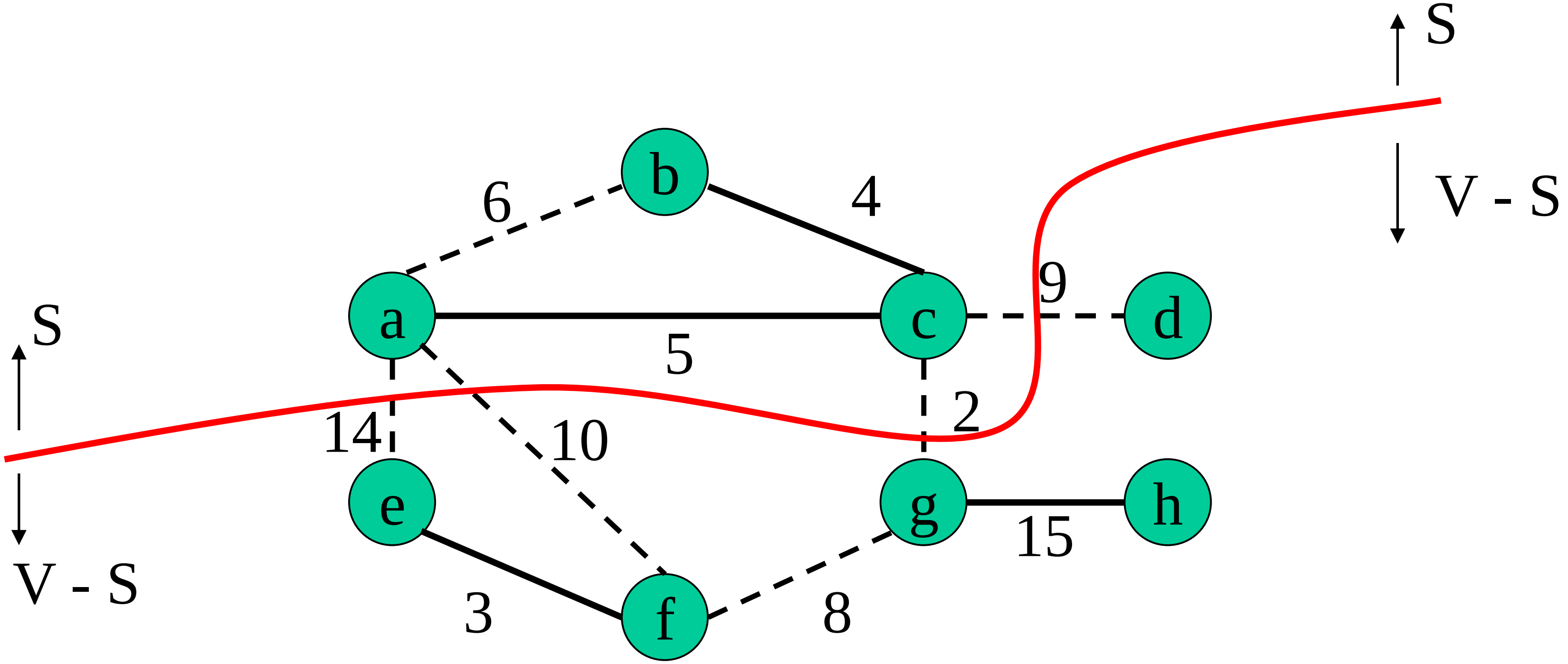
- Let A $\subset$ E be a subset of some MST T of G (i.e. A $\subset$ T)
- Let (S, V - S) be any cut of G that respects A
- Let $(u,v) \in$ E be a light edge crossing (S, V - S)

➢ Then, edge (u,v) is a safe edge for A

➢ Example:



Properties of Generic MST Algorithm

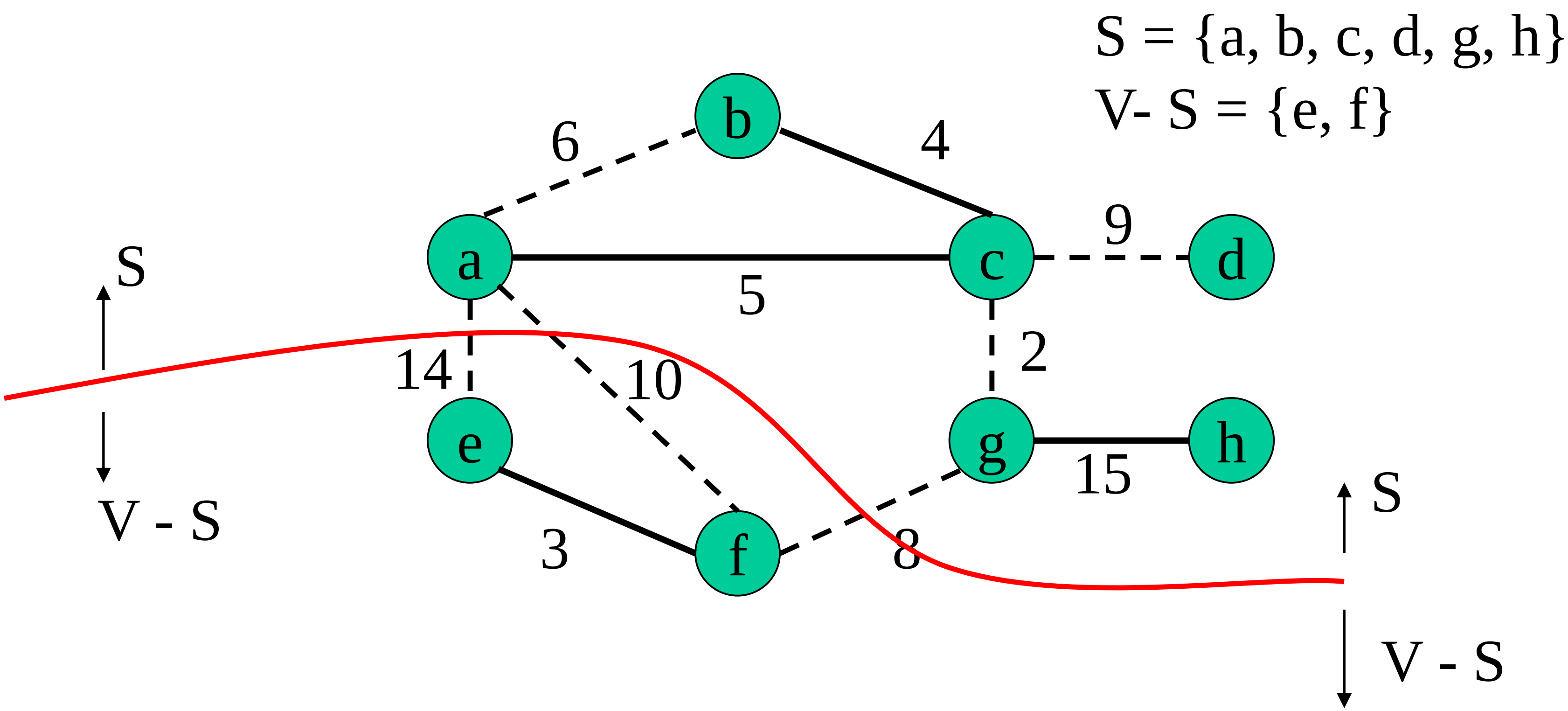


- S = { a, b, c}
- V – S = { d, e, f, g, h}
- A = { (a,c), (b,c), (e,f), (g,h) }
- CUT (S, V - S) repects A

- Edge (c,g) $\in$ E is the light edge crossing the cut
- Edge (c,g) is a safe edge for A

Properties of Generic MST Algorithm

Example : Consider another cut respecting the same A



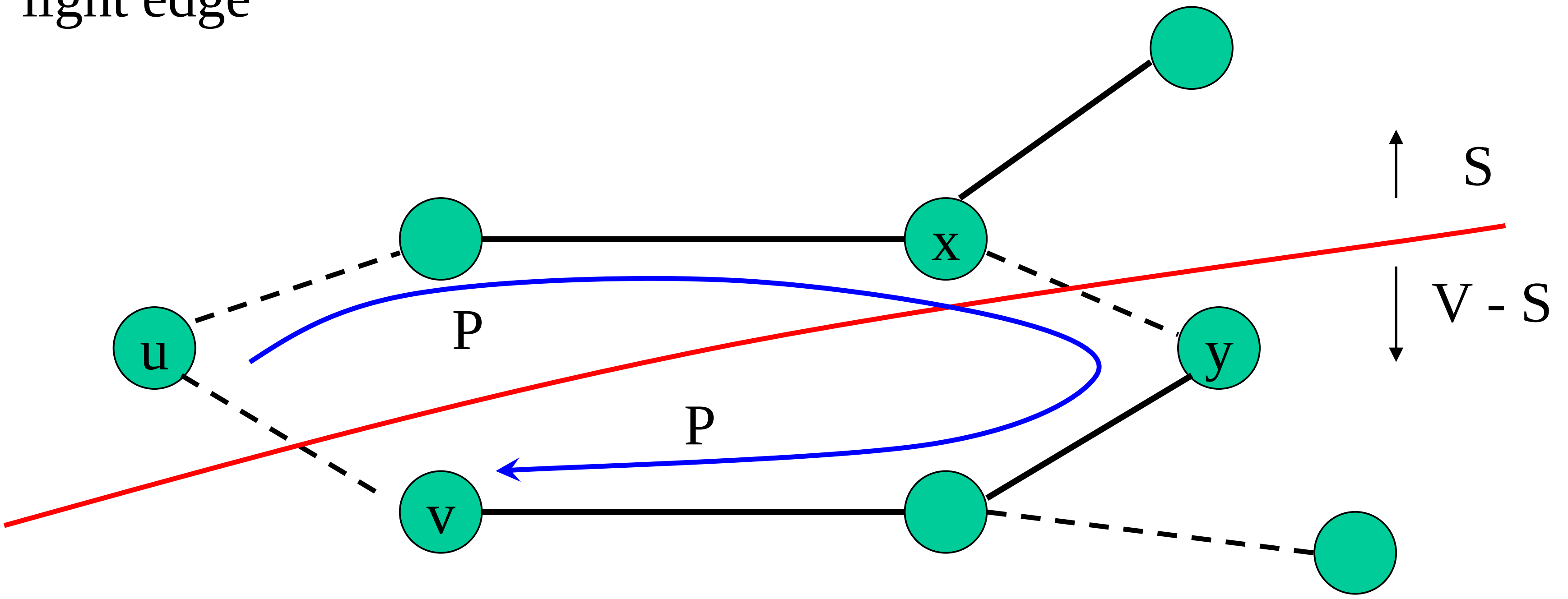
- Edge (g,f) is the light edge

Properties of Generic MST Algorithm

THM1 displays the greedy-choice property

Proof of THM1 :

- Consider an MST T $\ni$ A $\subset$ T and $(u,v) \notin$ T
- Try to construct another MST T' $\ni$ A U $\{(u,v)\} \subset$ T', (u,v) is a light edge



Properties of Generic MST Algorithm

- Consider the unique path P between *u* & *v* in T

- Since *u* & *v* are on opposite sides of the cut
  - $\exists$ at least one edge on P that crosses the cut
  - Let (*x*,*y*) be any such edge

- Removing (*x*,*y*) breaks T into two components T_1 & T_2
  - T_1 & T_2 are MST's of G_1 & G_2 (Optimal Substructure Prop.)

Properties of Generic MST Algorithm

- Therefore, $T' = T - \{(x,y)\} \cup \{(u,v)\}$ is also a spanning tree of G
- Since (u,v) is a light edge crossing the cut and (x,y) also crosses the cut

  - $\omega(u,v) \leq \omega(x,y)$

  - $\omega(T') = \omega(T) - \omega(x,y) + \omega(u,v) \leq \omega(T)$
- Since T is an MST, T' is also an MST □

Properties of Generic MST Algorithm

- Lemma: At any point in the execution of the generic algorithm
  - Graph $G_A = (V,A)$ is a forest
  - Each connected component of G_A is a tree
- Corollary: Let $A \subset E$ be a subtree of some MST T of G (i.e., $A \subset T$)

  Let C be a connected component in forest $G_A = (V,A)$
  - If (u,v) is a light edge connecting C to some other component in G_A then (u,v) is a safe edge for A
- Algorithms of Kruskal and Prim :
  - Both algorithms use a specific rule:
    - ➢ To determine a safe-edge in the Generic MST algoritm.

Properties of Generic MST Algorithm

- In Kruskal's algorithm
  - Set A : A forest
  - Safe-Edge : Least-Weight edge connecting two components (trees).
- In Prim's algorithm
  - Set A : A single tree
  - Safe-Edge : Least-Weight edge connecting tree to a vertex not in the tree.