Minimum Spanning Tree (MST)

Hengfeng Wei

hfwei@nju.edu.cn

June 19, 2018

Cut Property

$$G = (V, E, w)$$

Cut Property (Strong)

- X is some part of an MST T of G
- Any *cut* $(S, V \setminus S)$ *s.t.* X does *not* cross $(S, V \setminus S)$ Âŋ
- Let e be *a* lightest edge across $(S, V \setminus S)$

Cut Property (Strong)

- X is some part of an MST T of G
- Any *cut* $(S, V \setminus S)$ *s.t.* X does *not* cross $(S, V \setminus S)$ Âŋ
- Let e be *a* lightest edge across $(S, V \setminus S)$

Then $X \cup \{e\}$ is some part of an MST T' of G.

Cut Property (Strong)

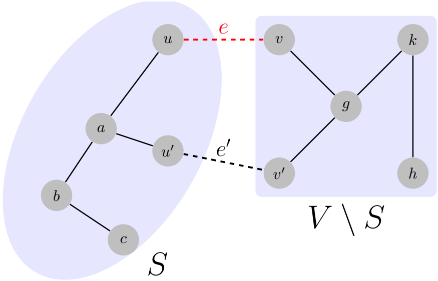
- X is some part of an MST T of G
- Any *cut* $(S, V \setminus S)$ *s.t.* X does *not* cross $(S, V \setminus S)$ Âŋ
- Let e be *a* lightest edge across $(S, V \setminus S)$

Then $X \cup \{e\}$ is some part of an MST T' of G.

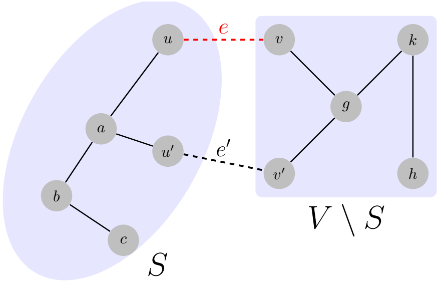
Correctness of Prim's and Kruskal's algorithms.

By Exchange Argument.

By Exchange Argument.



By Exchange Argument.



$$T + \{e\} - \{e'\}$$

Cut Property (Weak)

A cut $(S, V \setminus S)$

Let $e = (u, v)$ be *a* minimum-weight edge across $(S, V \setminus S)$

Cut Property (Weak)

A cut $(S, V \setminus S)$

Let $e = (u, v)$ be *a* minimum-weight edge across $(S, V \setminus S)$

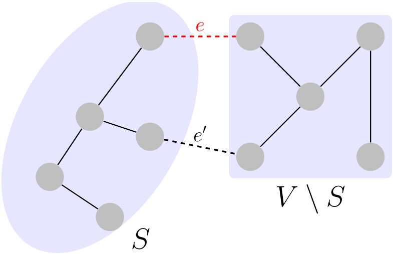
Then e must be in *some* MST of G.

Cut Property (Weak)

A cut $(S, V \setminus S)$

Let $e = (u, v)$ be *a* minimum-weight edge across $(S, V \setminus S)$

Then e must be in *some* MST of G.

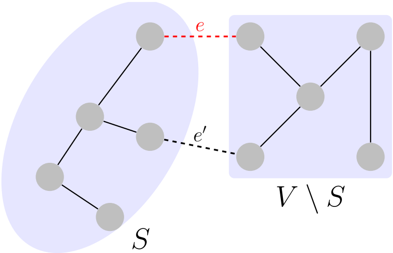


Cut Property (Weak)

A cut $(S, V \setminus S)$

Let $e = (u, v)$ be *a* minimum-weight edge across $(S, V \setminus S)$

Then e must be in *some* MST of G.



"a" → "the" $\Longrightarrow$ "some" → "any"

Converse of Cut Property (Weak)

$$e = (u, v) \in \exists \text{ MST } T \text{ of } G$$

$$\Longrightarrow$$

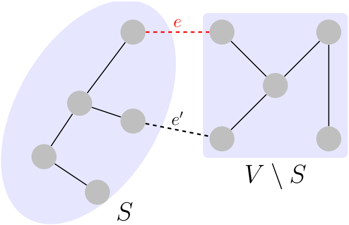
$$e \text{ is a lightest edge across some cut } (S, V \setminus S)$$

Converse of Cut Property (Weak)

$$e = (u, v) \in \exists \text{ MST } T \text{ of } G$$

$$\Longrightarrow$$

e is a lightest edge across some cut $(S, V \setminus S)$



$$T' = \underbrace{T - \{e\}}_{\text{to find } (S, V\setminus S)} + \underbrace{\{e'\}}_{\exists ?}$$

Application of Cut Property [Problem: 10.15 (3)]

$e \in G$ is a lightest edge $\implies$ $e \in \exists$ MST of G

Application of Cut Property [Problem: 10.15 (3)]

$e \in G$ is a lightest edge $\implies$ $e \in \exists$ MST of G

$$(S = \{u\}, V \setminus S)$$

Application of Cut Property [Problem: 10.15 (4)]

$$e \in G \text{ is the unique lightest edge} \Rightarrow e \in \forall \text{ MST}$$

Application of Cut Property [Problem: 10.15 (4)]

$e \in G$ is the unique lightest edge $\Rightarrow e \in \forall$ MST

By contradiction.

$$e \notin T : T' = T + \{e\} - \{e'\} \implies w(T') < w(T)$$

Wrong divide-and-conquer algorithm for MST [Problem: 10.21]

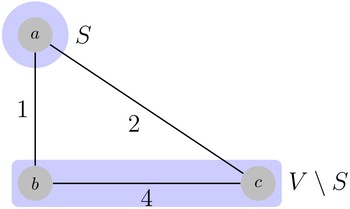
$$(V_1, V_2) : ||V_1| - |V_2|| \leq 1$$

$$T_1 + T_2 + \{e\} : e \text{ is a lightest edge across } (V_1, V_2)$$

Wrong divide-and-conquer algorithm for MST [Problem: 10.21]

$$(V_1, V_2) : ||V_1| - |V_2|| \leq 1$$

$$T_1 + T_2 + \{e\} : e \text{ is a lightest edge across } (V_1, V_2)$$



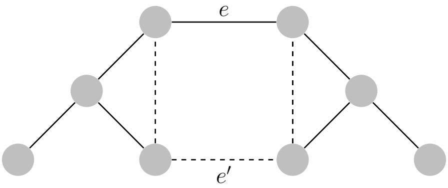
Cycle Property

$$G = (V, E, w)$$

Cycle property [Problem: $10.19(b)$]

- Let C be any cycle in G
- Let $e = (u, v)$ be *a* maximum-weight edge in C

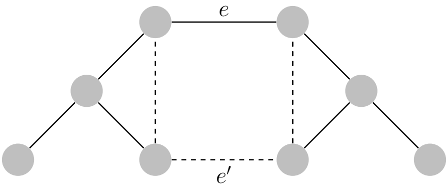
Then $\exists$ MST T of $G : e \notin T$.



Cycle property [Problem: $10.19(b)$]

- Let C be any cycle in G
- Let $e = (u, v)$ be *a* maximum-weight edge in C

Then $\exists$ MST T of $G : e \notin T$.



$$T' = \underbrace{T - \{e\}}_{e \in T} + \{e'\}$$

Anti-Kruskal algorithm [Problem: $10.19(c)$]

Reverse-delete algorithm (wiki)

$$O\Big(m \log n (\log \log n)^3\Big)$$

Anti-Kruskal algorithm [Problem: $10.19(c)$]

Reverse-delete algorithm (wiki)

$$O\Big(m \log n (\log \log n)^3\Big)$$

Proof.

Invariant: If F is the set of edges remained at the end of the while loop, then there is some MST that are a subset of F. □

Anti-Kruskal algorithm [Problem: $10.19(c)$]

Reverse-delete algorithm (wiki)

$$O\Big(m \log n (\log \log n)^3\Big)$$

Proof.

Invariant: If F is the set of edges remained at the end of the while loop, then there is some MST that are a subset of F. □

"On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem" — Kruskal, 1956.

Application of Cycle Property [Problem: 10.15 (1)]

$$G = (V, E), |E| > |V| - 1, \ e \text{ unique maximum-weighted edge}$$

$$\overset{?}{\Longrightarrow}$$

$$e \notin \text{any MST}$$

Application of Cycle Property [Problem: 10.15 (1)]

$$G = (V, E), |E| > |V| - 1, \; e \text{ unique maximum-weighted edge}$$

$$\overset{?}{\Longrightarrow}$$

$$e \notin \text{any MST}$$

Bridge

Application of Cycle Property [Problem: 10.15 (2)]

$C \subseteq G, e \in C$, e is the unique maximum-weighted edge of G

$$\Longrightarrow$$

$e \notin$ any MST of G

Application of Cycle Property [Problem: 10.15 (2)]

$$C \subseteq G, e \in C, e \text{ is the unique maximum-weighted edge of } G$$

$$\Longrightarrow$$

$$e \notin \text{any MST of } G$$

By contradiction.

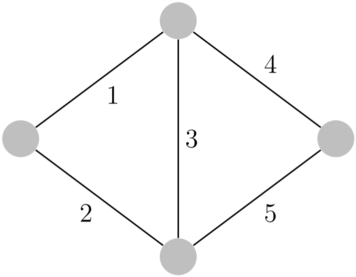
$$T' = T - \{e\} + \{e'\}$$

Application of Cycle Property [Problem: $10.15\ (5)$]

$C \subseteq G, e \in C$, e is the unique lightest edge of $C \implies {}^{?}e \in \forall$ MST

Application of Cycle Property [Problem: 10.15 (5)]

$C \subseteq G, e \in C$, e is the unique lightest edge of $C \implies {}^{?} e \in \forall$ MST



Uniqueness of MST [Problem: 10.18 (1)]

Distinct weights $\implies$ unique MST.

Uniqueness of MST [Problem: 10.18 (1)]

Distinct weights $\Longrightarrow$ unique MST.

By contradiction.

Uniqueness of MST [Problem: 10.18 (1)]

Distinct weights $\implies$ unique MST.

By contradiction.

$\exists$ MSTs $T_1 \neq T_2$

Uniqueness of MST [Problem: 10.18 (1)]

Distinct weights $\implies$ unique MST.

By contradiction.

$\exists$ MSTs $T_1 \neq T_2$

$$\Delta E = \{e \mid e \in T_1 \setminus T_2 \lor e \in T_2 \setminus T_1\}$$

Uniqueness of MST [Problem: 10.18 (1)]

Distinct weights $\implies$ unique MST.

By contradiction.

$$\exists \text{ MSTs } T_1 \neq T_2$$

$$\Delta E = \{e \mid e \in T_1 \setminus T_2 \lor e \in T_2 \setminus T_1\}$$

$$e = \min \Delta E$$

Uniqueness of MST [Problem: 10.18 (1)]

Distinct weights $\implies$ unique MST.

By contradiction.

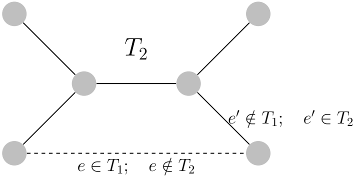
$\exists$ MSTs $T_1 \neq T_2$

$$\Delta E = \{e \mid e \in T_1 \setminus T_2 \lor e \in T_2 \setminus T_1\}$$

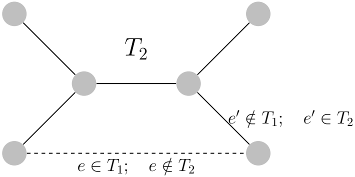
$$e = \min \Delta E$$

$e \in T_1 \setminus T_2$ *(w.l.o.g)*

$$e \in T_1 \setminus T_2$$

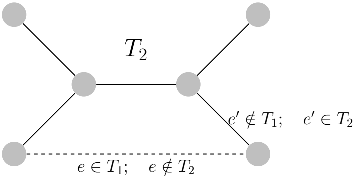

T_2
$e' \notin T_1; \quad e' \in T_2$
$e \in T_1; \quad e \notin T_2$

$$e \in T_1 \setminus T_2$$


T_2
$e' \notin T_1; \quad e' \in T_2$
$e \in T_1; \quad e \notin T_2$

$$T_2 + \{e\} \implies C$$

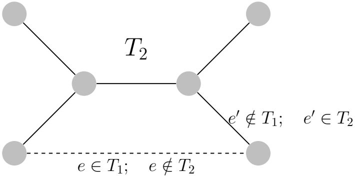
$$e \in T_1 \setminus T_2$$


T_2
$e' \notin T_1; \quad e' \in T_2$
$e \in T_1; \quad e \notin T_2$

$$T_2 + \{e\} \implies C$$

$$\exists (e' \in C) \notin T_1 \implies e' \in T_2 \setminus T_1 \implies e' \in \Delta E$$

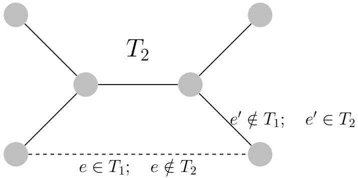
$$e \in T_1 \setminus T_2$$


T_2
$e' \notin T_1; \quad e' \in T_2$
$e \in T_1; \quad e \notin T_2$

$$T_2 + \{e\} \implies C$$

$$\exists (e' \in C) \notin T_1 \implies e' \in T_2 \setminus T_1 \implies e' \in \Delta E \implies w(e') > w(e)$$

$$e \in T_1 \setminus T_2$$


T_2
$e' \notin T_1; \quad e' \in T_2$
$e \in T_1; \quad e \notin T_2$

$$T_2 + \{e\} \implies C$$

$$\exists (e' \in C) \notin T_1 \implies e' \in T_2 \setminus T_1 \implies e' \in \Delta E \implies w(e') > w(e)$$

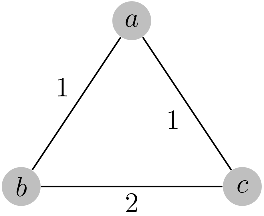
$$T' = T_2 + \{e\} - \{e'\} \implies w(T') < w(T_2)$$

Condition for Uniqueness of MST [Problem: 10.18 (2)]

Unique MST $\not\Rightarrow$ Equal weights.

Condition for Uniqueness of MST [Problem: 10.18 (2)]

$$\text{Unique MST} \not\Rightarrow \text{Equal weights.}$$

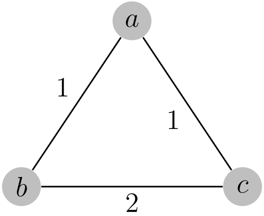


Unique MST [Problem: $10.21(3)$]

Unique MST $\not\Rightarrow$ Minimum-weight edge across any cut is unique.

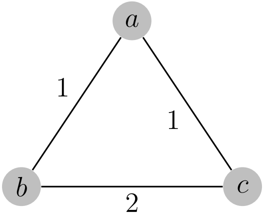
Unique MST [Problem: $10.21(3)$]

Unique MST $\not\Rightarrow$ Minimum-weight edge across any cut is unique.



Unique MST [Problem: 10.21(3)]

Unique MST $\not\Longrightarrow$ Minimum-weight edge across any cut is unique.



Theorem

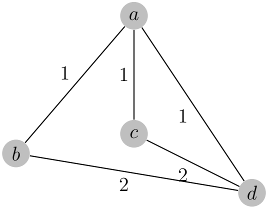
Minimum-weight edge across any cut is unique $\Longrightarrow$ *Unique MST.*

Unique MST [Problem: 10.21 (3)]

Unique MST $\not\Longrightarrow$ Maximum-weight edge in any cycle is unique.

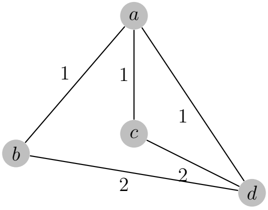
Unique MST [Problem: 10.21 (3)]

Unique MST $\not\Longrightarrow$ Maximum-weight edge in any cycle is unique.



Unique MST [Problem: 10.21 (3)]

Unique MST $\not\Longrightarrow$ Maximum-weight edge in any cycle is unique.

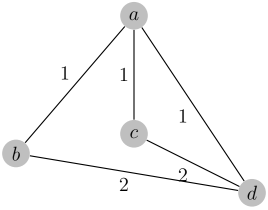


Theorem (Conjecture)

Maximum-weight edge in any cycle is unique $\Longrightarrow$ *Unique MST.*

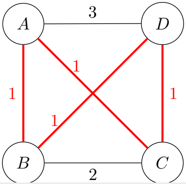
Unique MST [Problem: 10.21 (3)]

Unique MST $\not\Longrightarrow$ Maximum-weight edge in any cycle is unique.



Theorem (Conjecture)

Maximum-weight edge in any cycle is unique $\Longrightarrow$ *Unique MST.*



Unique MST [Problem: 10.21 (4)]

To decide whether a graph has a unique MST.

Unique MST [Problem: 10.21 (4)]

To decide whether a graph has a unique MST.

Ties breaking in Prim's and Kruskal's algorithms

Unique MST [Problem: 10.21 (4)]

To decide whether a graph has a unique MST.

Ties breaking in Prim's and Kruskal's algorithms

Proof.

Cut property and Cycle property. □

Adding a Vertex v to MST T [Problem: 10.7]

$$G' = (V', E') : V' = V + \{v\}, E' = E + E_v$$

To find an MST T' of G'.

$$O\Big((m+n)\log n\Big) \quad \text{(recompute on } G'\text{)}$$

Adding a Vertex v to MST T [Problem: 10.7]

$$G' = (V', E') : V' = V + \{v\}, E' = E + E_v$$

To find an MST T' of G'.

$$O\Big((m+n)\log n\Big) \quad \text{(recompute on } G'\text{)}$$

Theorem

There exists an MST of G' that includes no edges in $G \setminus T$.

Adding a Vertex v to MST T [Problem: 10.7]

$$G' = (V', E') : V' = V + \{v\}, E' = E + E_v$$

To find an MST T' of G'.

$$O\Big((m+n)\log n\Big) \quad \text{(recompute on } G'\text{)}$$

Theorem

There exists an MST of G' that includes no edges in $G \setminus T$.

$$O(n \log n) \quad \text{(recompute on } G'' = (V + \{v\}, T + E_v)\text{)}$$

Adding a Vertex v to MST T [Problem: 10.7]

$$G' = (V', E') : V' = V + \{v\}, E' = E + E_v$$

To find an MST T' of G'.

$$O\Big((m+n)\log n\Big) \quad \text{(recompute on } G'\text{)}$$

Theorem

There exists an MST of G' that includes no edges in $G \setminus T$.

$$O(n \log n) \quad \text{(recompute on } G'' = (V + \{v\}, T + E_v)\text{)}$$

$$O(n)$$

Adding a Vertex v to MST T [Problem: 10.7]

$$G' = (V', E') : V' = V + \{v\}, E' = E + E_v$$

To find an MST T' of G'.

$$O\Big((m+n)\log n\Big) \quad \text{(recompute on } G'\text{)}$$

Theorem

There exists an MST of G' that includes no edges in $G \setminus T$.

$$O(n \log n) \quad \text{(recompute on } G'' = (V + \{v\}, T + E_v)\text{)}$$

$$O(n)$$

"On Finding and Updating Spanning Tress and Shortest Paths", 1975
"Algorithms for Updating Minimum Spanning Trees", 1978

Feedback Edge Set: [Problem: 10.8]

1. Maximum spanning tree
2. (Minimum) feedback edge set:
   - a set of edges which, when removed from the graph, leave an acyclic graph G'
   - assuming G is connected $\Rightarrow$ G' is connected
   - feedback *arc* set: "cycle" $\Rightarrow$ circular dependency

- G' is connected $+$ acyclic $\Rightarrow$ G' is an ST
- $\text{FES} \Leftrightarrow G \setminus \text{Max-ST}$

Edge Weights

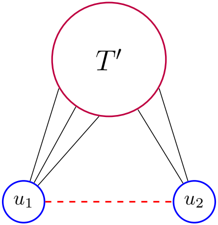
- [Problem: 10.15 (7)]: negative edges for Prim algorithm
- [Problem: 10.16]: $w'(e) = (w(e))^2$

MST with Specified Leaves: [Problem: 10.11]

- $G = (V, E), U \subset V$
- finding an MST with U as leaves

MST with Specified Leaves: [Problem: 10.11]

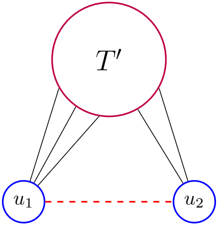
- $G = (V, E), U \subset V$
- finding an MST with U as leaves



MST T' of $G' = G \setminus U$

MST with Specified Leaves: [Problem: 10.11]

- $G = (V, E), U \subset V$
- finding an MST with U as leaves



MST T' of $G' = G \setminus U$

Attach $\forall u \in U$ to T' (with lightest edge)

MST with Specified Edges: [Problem: 10.13]

$$G = (V, E), \quad S \subset E \text{ (no cycle in } S\text{)}$$

To find an MST with S as edges.

MST with Specified Edges: [Problem: 10.13]

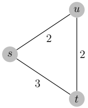
$$G = (V, E), \quad S \subset E \text{ (no cycle in } S\text{)}$$

To find an MST with S as edges.

$G \to G'$: contract each component of S to a vertex

MST vs. Shortest Paths [Problem: 10.15 (6)]

✗The shortest path between two nodes is necessarily part of some MST.



Sharing edges [Problem: 3.6.5]

- $G = (V, E), w(e) > 0$
- Given s: all sssp trees from s must share some edge with all (some) MSTs of G

✓$w > 0$; Vertex s; shortest-path tree of s and some MST share a common edge [Problem: 10.9]

Solution

E': lightest edges leaving s

- any MST T of G: $T \cap E' \neq \emptyset$
- $E' \subset \forall$ sssp trees

THANK YOU