3-5 Minimum Spanning Trees

Hengfeng Wei

hfwei@nju.edu.cn

October 22, 2018

[Problem: 4.8]

$$\forall v \in V(G) : \deg(v) \geq 2 \implies G \text{ contains a cycle}$$

Cut Property

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

Cut Property (I)

$$
\begin{aligned}
X &: \text{ A part of some MST } T \text{ of } G \\
(S, V \setminus S) &: \text{ A \textit{cut} such that } X \text{ does \textit{not} cross } (S, V \setminus S) \\
e &: \text{ \textit{A} lightest edge across } (S, V \setminus S)
\end{aligned}
$$

Cut Property (I)

$$X: \text{ A part of some MST } T \text{ of } G$$

$(S, V \setminus S)$: A *cut* such that X does *not* cross $(S, V \setminus S)$

e : *A* lightest edge across $(S, V \setminus S)$

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

Cut Property (I)

$$X: \text{A part of some MST } T \text{ of } G$$

$$(S, V \setminus S): \text{A } \textit{cut} \text{ such that } X \text{ does } \textit{not} \text{ cross } (S, V \setminus S)$$

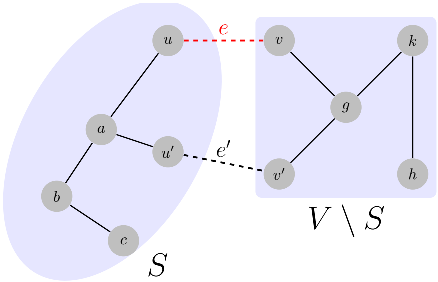
$$e: \textit{A} \text{ lightest edge across } (S, V \setminus S)$$

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

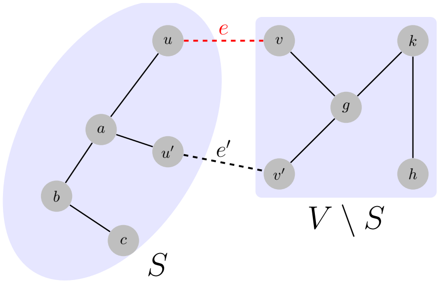
Correctness of Prim's and Kruskal's algorithms.

By Exchange Argument.

By Exchange Argument.

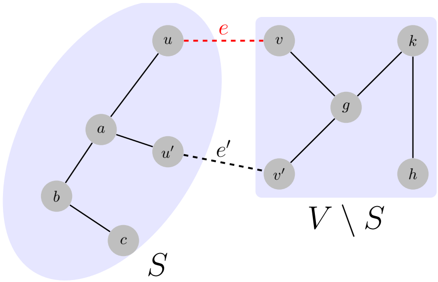


By Exchange Argument.



$$T' = \underbrace{T + \{e\}}_{\text{if } e \notin T} - \{e'\}$$

By Exchange Argument.



$$T' = \underbrace{T + \{e\}}_{\textcolor{red}{\text{if } e \notin T}} - \{e'\}$$

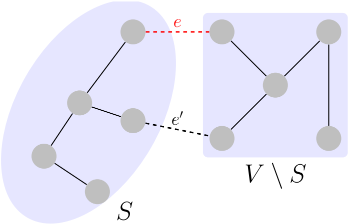
"a" $\to$ "the" $\Longrightarrow$ "some" $\to$ "all"

Cut Property (II)

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

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

$\exists$ MST T of $G : e \in T$

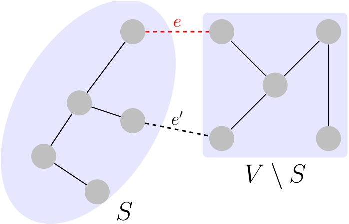


Cut Property (II)

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

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

$\exists$ MST T of $G : e \in T$



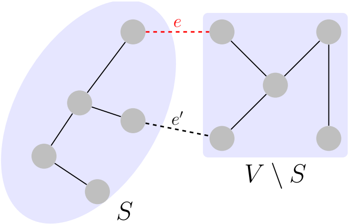
$$T' = \underbrace{T + \{e\}}_{\text{if } e \notin T} - \{e'\}$$

Cut Property (II)

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

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

$\exists$ MST T of $G : e \in T$



$$T' = \underbrace{T + \{e\}}_{\text{if } e \notin T} - \{e'\}$$

"a" $\to$ "the" $\Longrightarrow$ "$\exists$" $\to$ "$\forall$"

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

$$e = (u, v) \in G \text{ is a lightest edge } \implies e \in \exists \text{ MST of } G$$

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

$$e = (u, v) \in G \text{ is the unique lightest edge } \implies e \in \forall \text{ MST}$$

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

$e = (u, v) \in G$ is a lightest edge $\implies e \in \exists$ MST of G

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

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

$e = (u, v) \in G$ is the unique lightest edge $\implies e \in \forall$ MST

Wrong Divide&Conquer Algorithm for MST [Problem: 10.21]

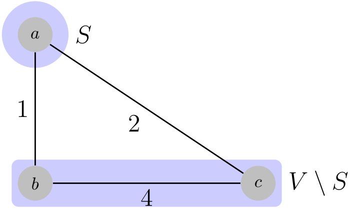
$$(V_1, V_2) : \Big| |V_1| - |V_2| \Big| \le 1$$

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

Wrong Divide&Conquer Algorithm for MST [Problem: 10.21]

$$(V_1, V_2) : \Big| |V_1| - |V_2| \Big| \le 1$$

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

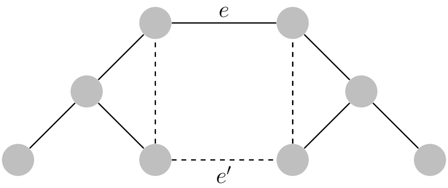


Cycle Property

Cycle Property [Problem: 10.19(b)]

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

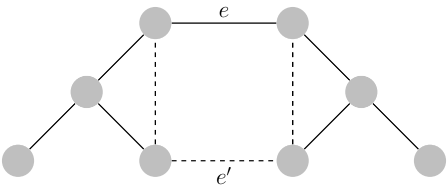
$$\text{Then } \exists \text{ MST } T \text{ of } G : e \notin T.$$



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 $G : e \notin T$.

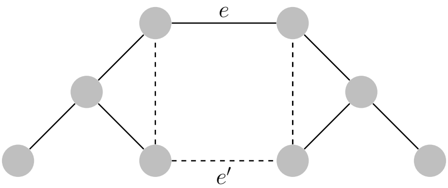


$$T' = \underbrace{T - \{e\}}_{\text{if } e \in T} + \{e'\}$$

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 $G : e \notin T$.



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

"a" $\to$ "the" $\Longrightarrow$ "$\exists$" $\to$ "$\forall$"

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

Reverse-delete algorithm (wiki; clickable)

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

Reverse-delete algorithm (wiki; clickable)

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

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

Reverse-delete algorithm (wiki; clickable)

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

Proof.

Cycle Property

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

Reverse-delete algorithm (wiki; clickable)

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

Proof.

Cycle Property

$$T \subseteq F \implies \exists\, T' : T' \subseteq F - \{e\}$$

□

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

Reverse-delete algorithm (wiki; clickable)

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

Proof.

Cycle Property

$$T \subseteq F \implies \exists\, T' : T' \subseteq F - \{e\}$$

□

"On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem"

— Kruskal, 1956.

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

$$G = (V, E), \quad |E| > |V| - 1$$

e : the unique maximum-weighted edge of G

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

$e \notin$ any MST

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

$$G = (V, E), \quad |E| > |V| - 1$$

e : the unique maximum-weighted edge of G

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

$e \notin$ any MST

Bridge

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

$$C \subseteq G, \quad e \in C$$

$$e: \text{ the unique maximum-weighted edge of } G$$

$$\Longrightarrow$$

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

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

$$C \subseteq G, \quad e \in C$$

$$e: \text{ the unique maximum-weighted edge of } G$$

$$\Longrightarrow$$

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

Cycle Property

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

$$C \subseteq G, \quad e \in C$$

$$e: \text{ the unique lightest edge of } C$$

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

$$e \in \forall \text{ MST}$$

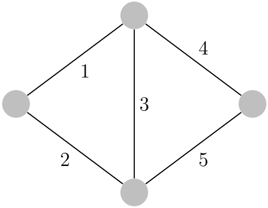
Application of Cycle Property [Problem: 10.15 (5)]

$$C \subseteq G, \quad e \in C$$

$$e: \text{ the unique lightest edge of } C$$

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

$$e \in \forall \text{ MST}$$



Uniqueness of MST

Uniqueness of MST [Problem: 10.18 (1)]

$$\text{Distinct weights} \implies \text{Unique MST.}$$

Uniqueness of MST [Problem: 10.18 (1)]

$$\text{Distinct weights} \implies \text{Unique MST.}$$

By Contradiction.

Uniqueness of MST [Problem: 10.18 (1)]

$$\text{Distinct weights} \implies \text{Unique MST.}$$

By Contradiction.

$\exists$ MSTs $T_1 \neq T_2$

Uniqueness of MST [Problem: 10.18 (1)]

Distinct weights $\implies$ Unique MST.

By Contradiction.

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

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

Uniqueness of MST [Problem: 10.18 (1)]

$$\text{Distinct weights} \implies \text{Unique MST.}$$

By Contradiction.

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

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

$$e = \min \Delta E$$

Uniqueness of MST [Problem: 10.18 (1)]

Distinct weights $\implies$ Unique MST.

By Contradiction.

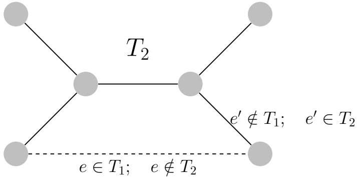
$$\exists \text{ MSTs } T_1 \neq T_2$$

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

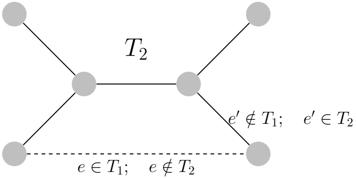
$$e = \min \Delta E$$

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

$$e \in T_1 \setminus T_2$$

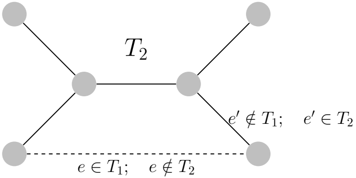


$$e \in T_1 \setminus T_2$$



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

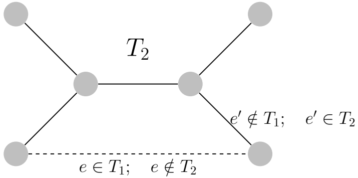
$$e \in T_1 \setminus T_2$$



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

$\exists (e' \in C) \notin T_1$

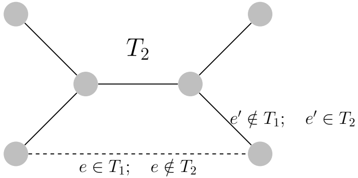
$$e \in T_1 \setminus T_2$$



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

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

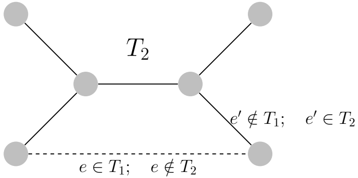
$$e \in T_1 \setminus T_2$$



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

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

$$e \in T_1 \setminus T_2$$



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

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

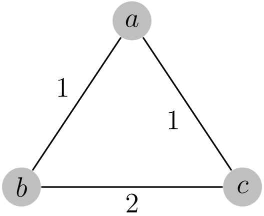
$$T' = T_2 + \{e\} - \{e'\} \implies w(T') < w(T_2)$$

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

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

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

$$\text{Unique MST} \;\not\Longrightarrow\; \text{Equal weights.}$$

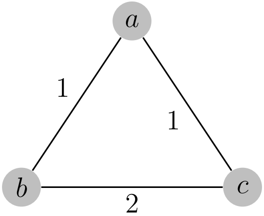


Unique MST [Problem: 10.21 (3)]

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

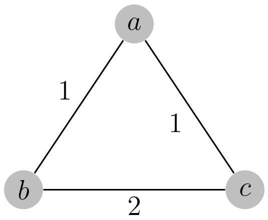
Unique MST [Problem: 10.21 (3)]

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 (After-class Exercise)

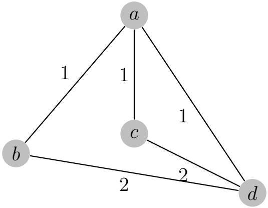
Minimum-weight edge across any cut is unique $\Longrightarrow$ *Unique MST.*

Unique MST [Problem: 10.21 (3)]

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

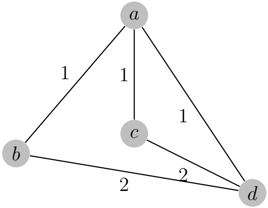
Unique MST [Problem: 10.21 (3)]

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



Unique MST [Problem: 10.21 (3)]

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



Theorem (After-class Exercise)

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)]

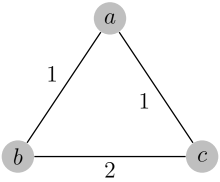
To decide whether a graph has a unique MST.

Ties in Prim's and Kruskal's algorithms

Unique MST [Problem: 10.21 (4)]

To decide whether a graph has a unique MST.

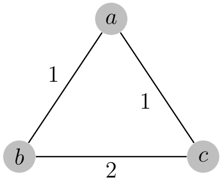
Ties in Prim's and Kruskal's algorithms



Unique MST [Problem: 10.21 (4)]

To decide whether a graph has a unique MST.

Ties in Prim's and Kruskal's algorithms

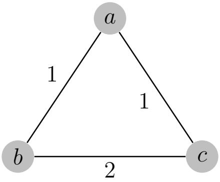


$$\underbrace{T}_{\text{Any MST}} + \underbrace{\{e\}, \forall e \notin T}_{\text{Cycle}}$$

Unique MST [Problem: 10.21 (4)]

To decide whether a graph has a unique MST.

Ties in Prim's and Kruskal's algorithms



$$\underbrace{T}_{\text{Any MST}} + \underbrace{\{e\}, \forall e \notin T}_{\text{Cycle}}$$

By Kruskal Algorithm.

Critical Edges [Problem: 10.12]

$$G \to T, \qquad G' \triangleq G \setminus \{e\} \to T'$$

$$w(T') > w(T)$$

Critical Edges [Problem: 10.12]

$$G \to T, \qquad G' \triangleq G \setminus \{e\} \to T'$$

$$w(T') > w(T)$$

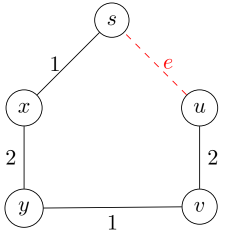
To find all critical edges in $O(m \log m)$ time.

Critical Edges [Problem: 10.12]

$$G \to T, \qquad G' \triangleq G \setminus \{e\} \to T'$$

$$w(T') > w(T)$$

To find all critical edges in $O(m \log m)$ time.



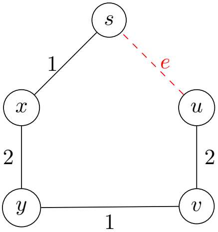
$w(e) = 3 \quad w(e) = 2$

Critical Edges [Problem: 10.12]

$$G \to T, \qquad G' \triangleq G \setminus \{e\} \to T'$$

$$w(T') > w(T)$$

To find all critical edges in $O(m \log m)$ time.



By Kruskal Algorithm.

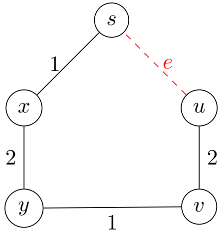
$w(e) = 3 \quad w(e) = 2$

Critical Edges [Problem: 10.12]

$$G \to T, \qquad G' \triangleq G \setminus \{e\} \to T'$$

$$w(T') > w(T)$$

To find all critical edges in $O(m \log m)$ time.



$w(e) = 3 \quad w(e) = 2$

By Kruskal Algorithm.

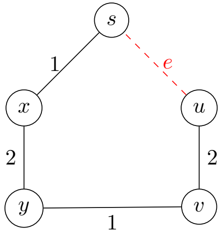
No missing: Check all cycles.

Critical Edges [Problem: 10.12]

$$G \to T, \qquad G' \triangleq G \setminus \{e\} \to T'$$

$$w(T') > w(T)$$

To find all critical edges in $O(m \log m)$ time.



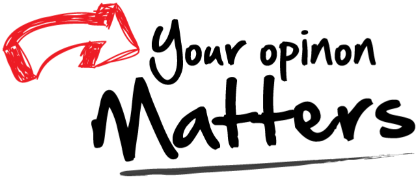
$w(e) = 3 \quad w(e) = 2$

By Kruskal Algorithm.

No missing: Check all cycles.

$O(m \log m)$


THANK YOU



Office 302

Mailbox: H016

hfwei@nju.edu.cn