3-5 Minimum Spanning Trees

Hengfeng Wei

hfwei@nju.edu.cn

October 22, 2018

Existence of Cycle (Problem 4.8)

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

Existence of Cycle (Problem 4.8)

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

By Contradiction.

Existence of Cycle (Problem 4.8)

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

By Contradiction.

$$m = n - k(G) \leq n - 1$$

Existence of Cycle (Problem 4.8)

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

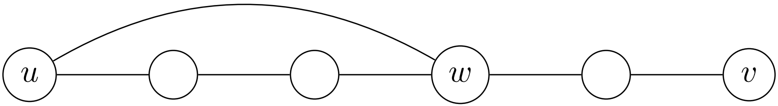
By Contradiction.

$$m = n - k(G) \leq n - 1$$

$$\sum_{v \in V(G)} \deg(v) \leq 2(n-1)$$

Existence of Cycle (Problem 4.8)

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



maximal path $P_{u,v}$

Existence of Cycle (Problem 4.8)

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

$\forall v \in V(G) : \deg(v) \geq 2$

Existence of Cycle (Problem 4.8)

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

$$\forall v \in V(G) : \deg(v) \geq 2$$

$$\implies \nexists v \in V(G) : \deg(v) = 1$$

Existence of Cycle (Problem 4.8)

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

$$\forall v \in V(G) : \deg(v) \geq 2$$

$$\implies \nexists v \in V(G) : \deg(v) = 1$$

$$\xRightarrow{\text{Theorem 4.3}} G \text{ is not a tree}$$

Existence of Cycle (Problem 4.8)

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

$$\forall v \in V(G) : \deg(v) \geq 2$$

$$\implies \nexists v \in V(G) : \deg(v) = 1$$

$$\xRightarrow{\text{Theorem 4.3}} G \text{ is not a tree}$$

$$\xRightarrow{\text{Theorem 4.2}} \exists u, v \in V(G) : u, v \text{ are connected by} \geq 2 \text{ paths}$$

Existence of Cycle (Problem 4.8)

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

$$\forall v \in V(G) : \deg(v) \geq 2$$

$$\implies \nexists v \in V(G) : \deg(v) = 1$$

$$\xRightarrow{\text{Theorem 4.3}} G \text{ is not a tree}$$

$$\xRightarrow{\text{Theorem 4.2}} \exists u, v \in V(G) : u, v \text{ are connected by} \geq 2 \text{ paths}$$

$$\implies G \text{ contains a cycle}$$

Existence of Cycle (Problem 4.8)

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

$$\forall v \in V(G) : \deg(v) \geq 2$$

$$\implies \nexists v \in V(G) : \deg(v) = 1$$

$$\xRightarrow{\text{Theorem 4.3}} G \text{ is not a tree}$$

$$\xRightarrow{\text{Theorem 4.2}} \exists u, v \in V(G) : u, v \text{ are connected by } \geq 2 \text{ paths}$$

$$\implies G \text{ contains a cycle}$$

Existence of Cycle (Problem 4.8)

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

$$\forall v \in V(G) : \deg(v) \geq 2$$

$$\implies \nexists v \in V(G) : \deg(v) = 1$$

$$\xRightarrow{\text{Theorem 4.3}} G \text{ is not a tree}$$

Consider each component G' of G

$$\xRightarrow{\text{Theorem 4.2}} \exists u, v \in V(G') : u, v \text{ are connected by } \geq 2 \text{ paths}$$

$$\implies G' \text{ contains a cycle}$$

Existence of Cycle (Problem 4.8)

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

$\forall v \in V(G) : \deg(v) \geq 2$

$\implies \nexists v \in V(G) : \deg(v) = 1$

$\xRightarrow{\text{Theorem 4.3}}$ G is not a tree

Consider each component G' of G

$\xRightarrow{\text{Theorem 4.2}}$ $\exists u, v \in V(G') : u, v$ are connected by ≥ 2 paths

$\implies G'$ contains a cycle

$\implies G$ contains a cycle

Bridge and Spanning Trees (Problem 4.26)

$$G: \text{ a connected graph}$$

$$e \in E(G) \text{ is a bridge} \iff e \in \forall \text{ST of } G$$

Bridge and Spanning Trees (Problem 4.26)

$$G: \text{ a connected graph}$$

$$e \in E(G) \text{ is a bridge} \iff e \in \forall \text{ST of } G$$

$$\text{“} \Longleftarrow \text{”}$$

Bridge and Spanning Trees (Problem 4.26)

$$G: \text{ a connected graph}$$

$$e \in E(G) \text{ is a bridge } \iff e \in \forall \text{ST of } G$$

" $\Longleftarrow$ "

By Contradiction.

Bridge and Spanning Trees (Problem 4.26)

$$G: \text{ a connected graph}$$

$$e \in E(G) \text{ is a bridge} \iff e \in \forall \text{ST of } G$$

"$\Longleftarrow$"

By Contradiction.

ST of $G - e$

Bridge and Spanning Trees (Problem 4.26)

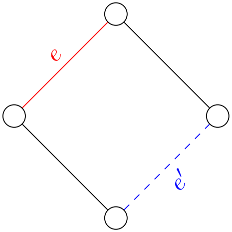
$$G: \text{ a connected graph}$$

$$e \in E(G) \text{ is a bridge } \iff e \in \forall \text{ST of } G$$

" $\Longleftarrow$ "

By Contradiction.

ST of $G - e$



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

Cut Property

Cut Property (Version I)

X : A part of some MST T of G

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

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

Cut Property (Version I)

X : A part of some MST T 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 (Version I)

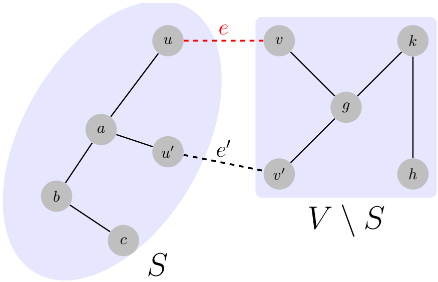
$$\begin{aligned} X &: \text{ A part of some MST } T \text{ of } G \\ (S, V \setminus S) &: \text{ A \textit{cut} such that } X \text{ does \textit{not} cross } (S, V \setminus S) \\ e &: \text{ A lightest edge across } (S, V \setminus S) \end{aligned}$$

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

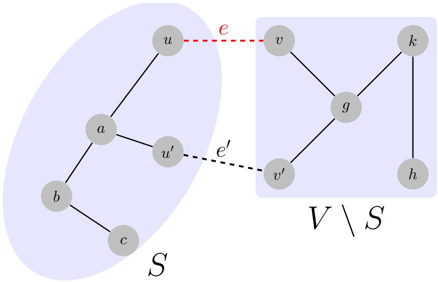
Correctness of Prim's and Kruskal's algorithms.

By Exchange Argument.

By Exchange Argument.

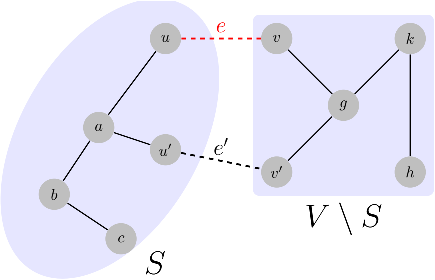


By Exchange Argument.



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

By Exchange Argument.



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

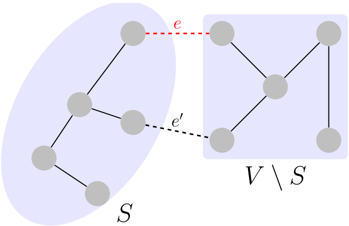
"a" $\to$ "the" $\implies$ "some" $\to$ "all"

Cut Property (Version II)

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

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

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

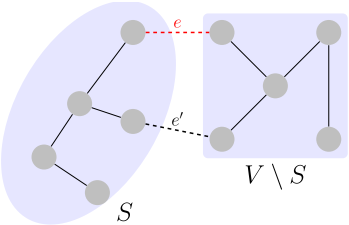


Cut Property (Version II)

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

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

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



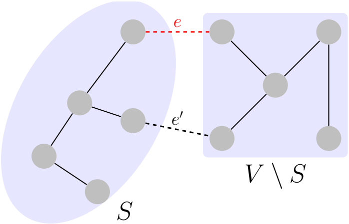
$$T' = \underbrace{T + \{e\}}_{\text{if } e \notin T} - \{e'\}$$

Cut Property (Version 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

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

Application of Cut Property

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

Application of Cut Property

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

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

Application of Cut Property

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

Wrong Divide&Conquer Algorithm for MST

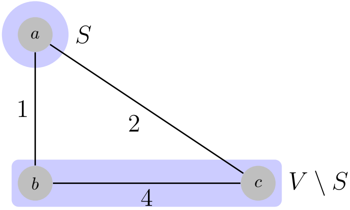
$$(V_1, V_2) : \Big| |V_1| - |V_2| \Big| \leq 1$$

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

Wrong Divide&Conquer Algorithm for MST

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

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

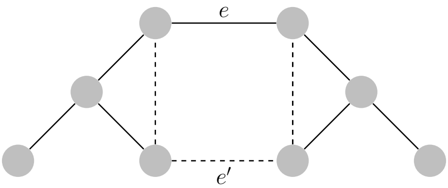


Cycle Property

Cycle Property

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

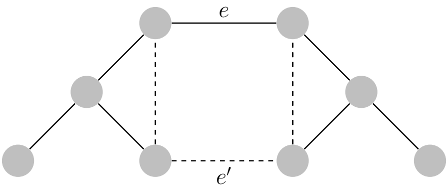
$$\text{Then } \exists \text{ MST } T \text{ of } G : e \notin T.$$



Cycle Property

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

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

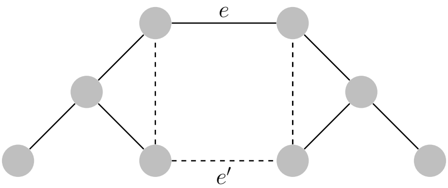


$$T' = \underbrace{T - \{e\}}_{\text{if } e \in T} + \{e'\}$$

Cycle Property

- 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

Reverse-delete algorithm (wiki; clickable)

Anti-Kruskal Algorithm

Reverse-delete algorithm (wiki; clickable)

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

Anti-Kruskal Algorithm

Reverse-delete algorithm (wiki; clickable)

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

Proof.

Cycle Property

Anti-Kruskal Algorithm

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

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

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

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

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

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

Application of Cycle Property

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

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

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

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

Bridge

Application of Cycle Property

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

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

$$\Longrightarrow$$

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

Application of Cycle Property

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

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

$$\implies$$

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

Cycle Property

Application of Cycle Property

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

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

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

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

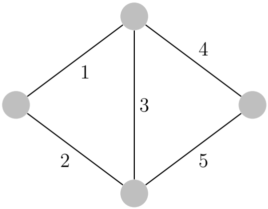
Application of Cycle Property

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

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

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

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



Uniqueness of MST

Uniqueness of MST (Problem 4.29)

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

Uniqueness of MST (Problem 4.29)

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

By Contradiction.

Uniqueness of MST (Problem 4.29)

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

By Contradiction.

$\exists$ MSTs $T_1 \neq T_2$

Uniqueness of MST (Problem 4.29)

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 4.29)

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

By Contradiction.

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

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

$$e = \min \Delta E$$

Uniqueness of MST (Problem 4.29)

Distinct weights $\implies$ Unique MST.

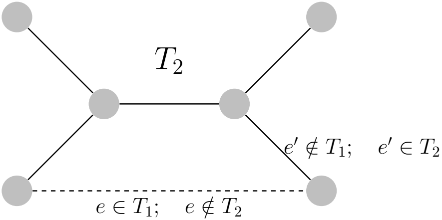
By Contradiction.

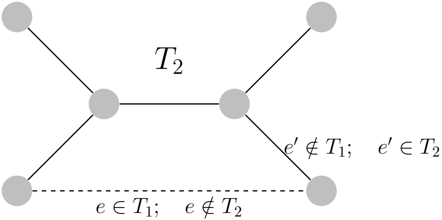
$$\exists \text{ MSTs } T_1 \neq T_2$$

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

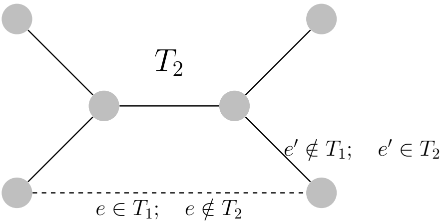
$$e = \min \Delta E$$

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


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

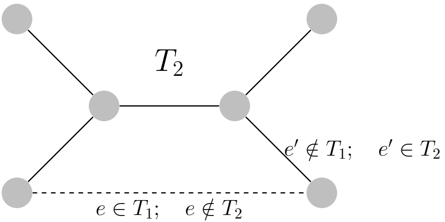


$$T_2 + \{e\} \implies C$$



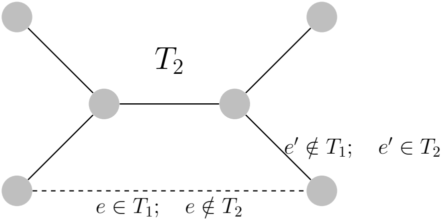
$$T_2 + \{e\} \implies C$$

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



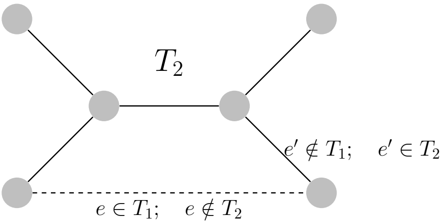
$$T_2 + \{e\} \implies C$$

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



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

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



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

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

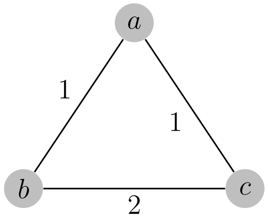
$$T' = T_2 + \{e\} - \{e'\} \implies w(T') < w(T_2)$$

Condition for Uniqueness of MST

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

Condition for Uniqueness of MST

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

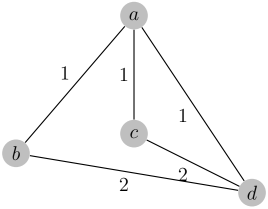


Unique MST (Problem 4.30)

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

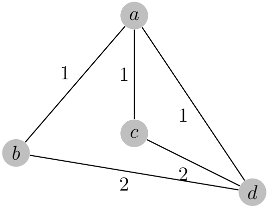
Unique MST (Problem 4.30)

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



Unique MST (Problem 4.30)

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



Theorem (After-class Exercise)

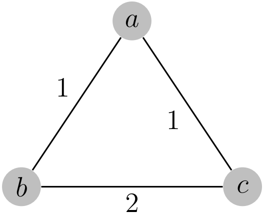
Maximum-weight edge in any cycle is unique $\Longrightarrow$ *Unique MST.*

Unique MST

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

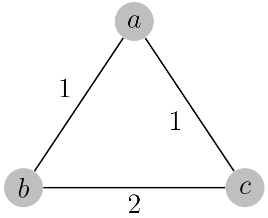
Unique MST

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



Unique MST

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

To decide whether a graph has a unique MST.

Unique MST

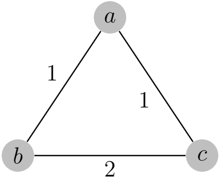
To decide whether a graph has a unique MST.

Ties in Prim's and Kruskal's algorithms

Unique MST

To decide whether a graph has a unique MST.

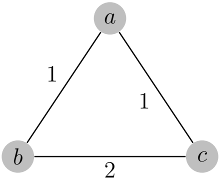
Ties in Prim's and Kruskal's algorithms



Unique MST

To decide whether a graph has a unique MST.

Ties in Prim's and Kruskal's algorithms

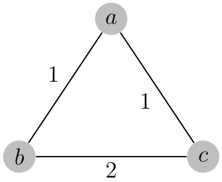


$$\underbrace{\color{red}{T}}_{\text{Any MST}} + \underbrace{\color{red}{\{e\}}, \forall e \notin T}_{\text{Cycle}}$$

Unique MST

To decide whether a graph has a unique MST.

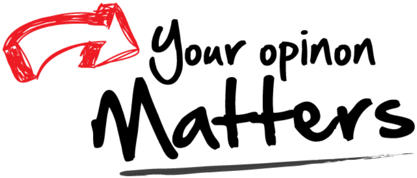
Ties in Prim's and Kruskal's algorithms



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

By Kruskal Algorithm.


THANK YOU



Office 302

Mailbox: H016

hfwei@nju.edu.cn