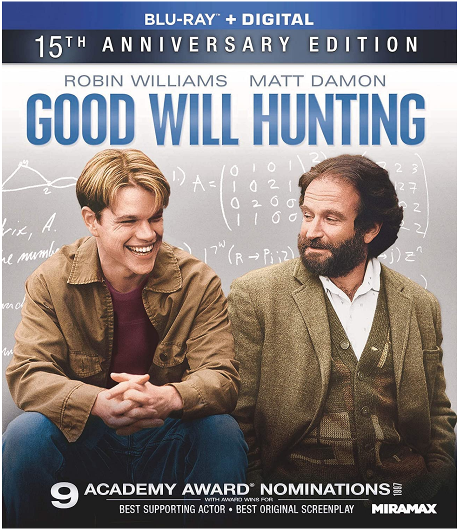
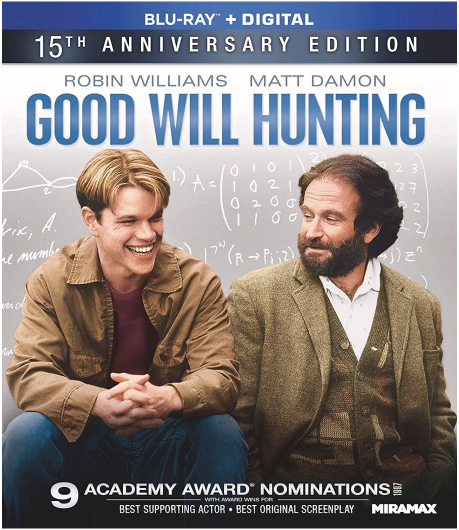
(十) 图论: 树 (Trees)

魏恒峰

hfwei@nju.edu.cn

2021 年 05 月 13 日


BLU-RAY™ + DIGITAL
15TH ANNIVERSARY EDITION
ROBIN WILLIAMS MATT DAMON
GOOD WILL HUNTING
9 ACADEMY AWARD® NOMINATIONS 1997
WITH AWARD WINS FOR
BEST SUPPORTING ACTOR • BEST ORIGINAL SCREENPLAY
MIRAMAX


BLU-RAY™ + DIGITAL
15TH ANNIVERSARY EDITION
ROBIN WILLIAMS MATT DAMON
GOOD WILL HUNTING
9 ACADEMY AWARD® NOMINATIONS 1997
WITH AWARD WINS FOR
BEST SUPPORTING ACTOR • BEST ORIGINAL SCREENPLAY
MIRAMAX

你, 真得, 看懂了吗?

Definition (Tree (树))

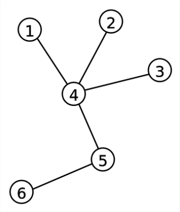
A tree is a connected acyclic undirected graph.

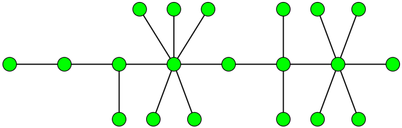
Definition (Tree (树))

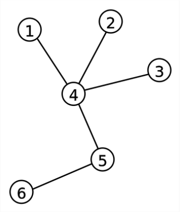
A tree is a connected acyclic undirected graph.

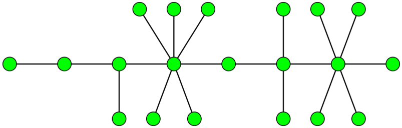
Definition (Forest (森林))

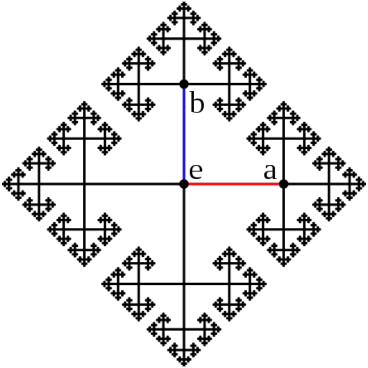
A forest is a acyclic undirected graph.

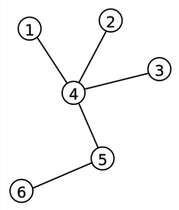

1
2
3
4
5
6

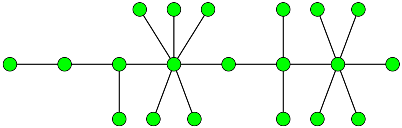


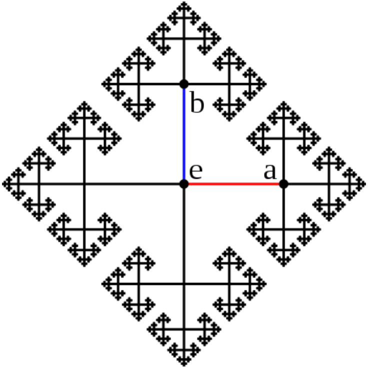

1
2
3
4
5
6




b
e
a


1
2
3
4
5
6




b
e
a

Definition (Internal Vertex (内部顶点); Leaf (叶子))

In a tree T with ≥ 2 vertices, for a vertex v in T, if

$$\deg(v) = 1$$

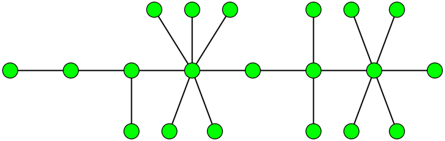
then v is called a leaf; otherwise, v is an internal vertex.

Definition (Internal Vertex (内部顶点); Leaf (叶子))

In a tree T with ≥ 2 vertices, for a vertex v in T, if

$$\deg(v) = 1$$

then v is called a leaf; otherwise, v is an internal vertex.

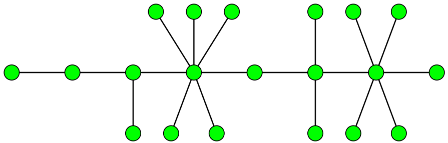


Definition (Internal Vertex (内部顶点); Leaf (叶子))

In a tree T with ≥ 2 vertices, for a vertex v in T, if

$$\deg(v) = 1$$

then v is called a leaf; otherwise, v is an internal vertex.



Lemma

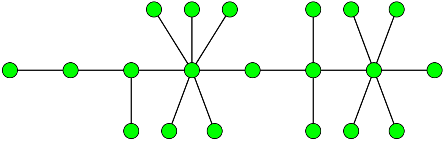
Any tree T with ≥ 2 vertices contains ≥ 1 leaf.

Definition (Internal Vertex (内部顶点); Leaf (叶子))

In a tree T with ≥ 2 vertices, for a vertex v in T, if

$$\deg(v) = 1$$

then v is called a leaf; otherwise, v is an internal vertex.



Lemma

Any tree T with ≥ 2 vertices contains ≥ 1 leaf.

$$\text{Otherwise, } \forall v \in V.\ \deg(v) \geq 2 \implies T \text{ has cycles.}$$

Lemma

Any tree T with ≥ 2 vertices contains ≥ 2 leaves.

Lemma

Any tree T with ≥ 2 vertices contains ≥ 2 leaves.

$$\sum_{v \in V} \deg(v) = 2n - 2$$

Lemma

Any tree T with ≥ 2 vertices contains ≥ 2 leaves.

$$\sum_{v \in V} \deg(v) = 2n - 2$$

Consider the two endpoints of any maximal (nontrivial) path in T.

Lemma

Any tree T with ≥ 2 vertices contains ≥ 2 leaves.

$$\sum_{v \in V} \deg(v) = 2n - 2$$

Consider the two endpoints of any maximal (nontrivial) path in T.

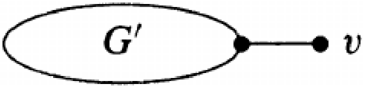
They are leaves of T.

Lemma

Deleting a leaf from a tree T with n vertices produces a tree with $n-1$ vertices.

Lemma

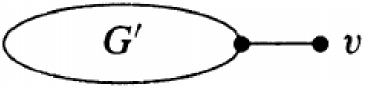
Deleting a leaf from a tree T with n vertices produces a tree with $n-1$ vertices.



$G' = G - v$ is connected and acyclic.

Lemma

Deleting a leaf from a tree T with n vertices produces a tree with $n-1$ vertices.



$G' = G - v$ is connected and acyclic.

A leaf does *not* belong to any paths connecting two other vertices.

Definition (Irreducible Tree)

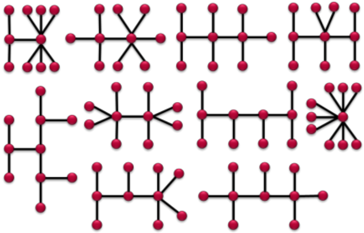
An irreducible tree is a tree T where

$$\forall v \in V(T).\ \deg(v) \neq 2.$$

Definition (Irreducible Tree)

An irreducible tree is a tree T where

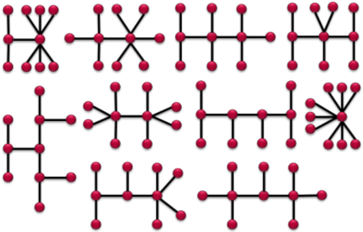
$$\forall v \in V(T).\ \deg(v) \neq 2.$$

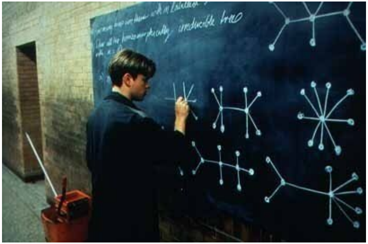


Definition (Irreducible Tree)

An irreducible tree is a tree T where

$$\forall v \in V(T).\ \deg(v) \neq 2.$$

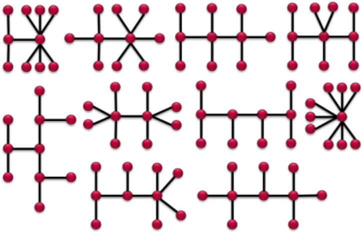


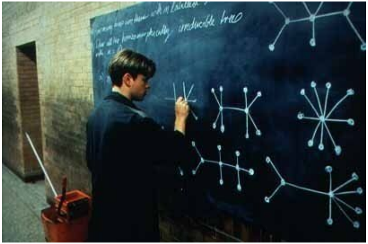


Definition (Irreducible Tree)

An irreducible tree is a tree T where

$$\forall v \in V(T).\ \deg(v) \neq 2.$$





Homeomorphically Irreducible Trees of size $n = 10$

Theorem ((We call it) Tree Theorem)

Let T be an undirected graph with n vertices.

Then the following statements are equivalent:

(1) *T is a tree;*

(2) *T is acyclic, and has $n-1$ edges;*

(3) *T is connected, and has $n-1$ edges;*

(4) *T is connected, and each edge is a bridge;*

(5) *Any two vertices of T are connected by exactly one path;*

(6) *T is acyclic, but the addition of any edge creates exactly one cycle.*

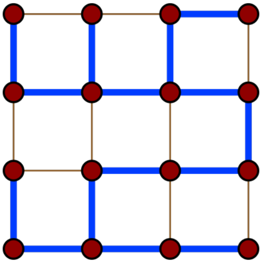

KEEP
CALM
AND
PROVE IT
KeepCalmAndPosters.com

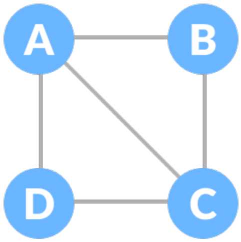
Definition (Spanning Tree (生成树))

A spanning tree T of an undirected graph G is a subgraph that is a tree with all vertices of G.

Definition (Spanning Tree (生成树))

A spanning tree T of an undirected graph G is a subgraph that is a tree with all vertices of G.





Definition (Subgraph (子图))

Definition (Subgraph (子图))

Definition (Induced Subgraph (诱导子图))

Theorem

Every connected undirected graph G admits a spanning tree.

Theorem

Every connected undirected graph G admits a spanning tree.

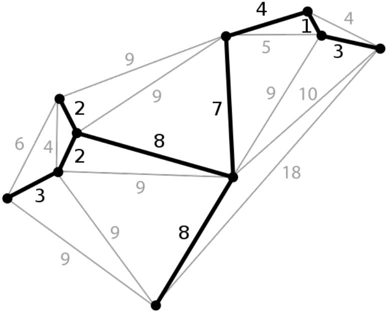
Repeatedly deleting vertices in cycles until the graph is acyclic.

Definition (Minimum Spanning Tree (MST; 最小生成树))

A minimum spanning tree T of an edge-weighted undirected graph G is a spanning tree with minimum total weight of edges.

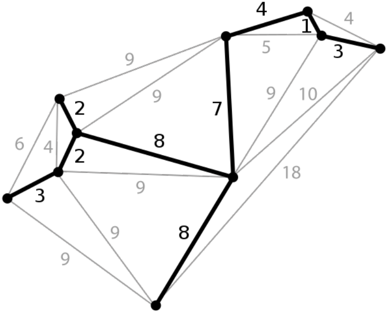
Definition (Minimum Spanning Tree (MST; 最小生成树))

A minimum spanning tree T of an edge-weighted undirected graph G is a spanning tree with minimum total weight of edges.



Definition (Minimum Spanning Tree (MST; 最小生成树))

A minimum spanning tree T of an edge-weighted undirected graph G is a spanning tree with minimum total weight of edges.



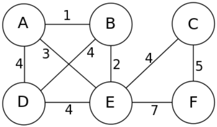
Existence? Uniqueness? Algorithms?

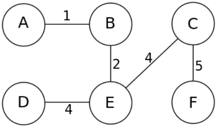
Theorem

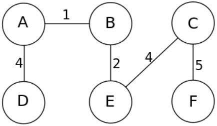
Every weighted connected undirected graph G admits a minimum spanning tree.

Theorem

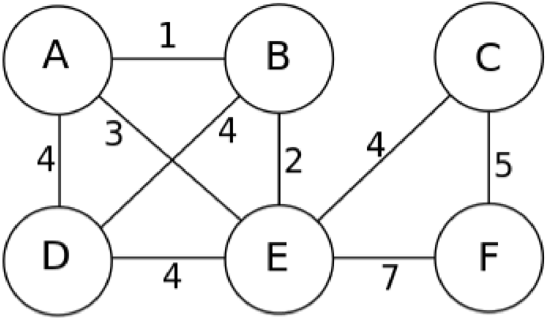
Every weighted connected undirected graph G admits a minimum spanning tree.



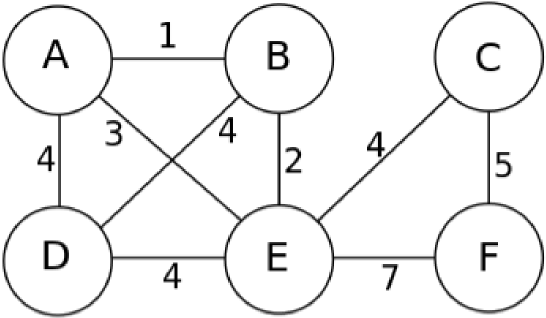




Joseph Kruskal (1928 ~ 2010)


A
B
C
D
E
F
1
4
3
4
2
4
5
4
7

Robert C. Prim (1921 ∼)


A
B
C
D
E
F
1
4
3
4
2
4
5
4
7

Cut Property

Cut Property (Version I)

X : A part of some MST T_1 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_1 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_2 of G.

Cut Property (Version I)

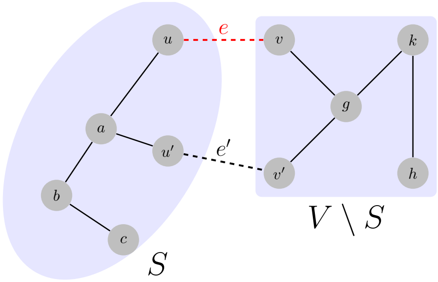
$$
\begin{aligned}
X &: \text{ A part of some MST } T_1 \text{ of } G \\
(S, V \setminus S) &: \text{ A \textit{cut} such that } X \text{ does \textit{not} cross } (S, V \setminus S) \\
e &: \text{ A lightest edge across } (S, V \setminus S)
\end{aligned}
$$

Then $X \cup \{e\}$ is a part of some MST T_2 of G.

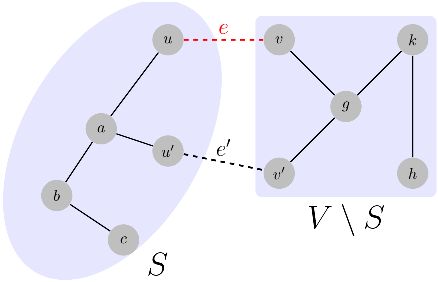
Correctness of Prim's and Kruskal's algorithms.

By Exchange Argument.

By Exchange Argument.

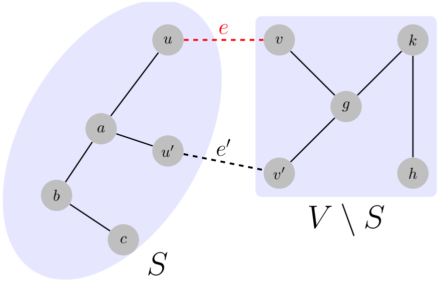


By Exchange Argument.



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

By Exchange Argument.



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

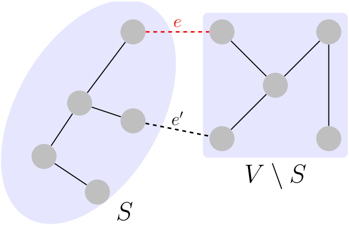
"a" $\rightarrow$ "the" $\Longrightarrow$ "some" $\rightarrow$ "all"

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 $G : e \in T$

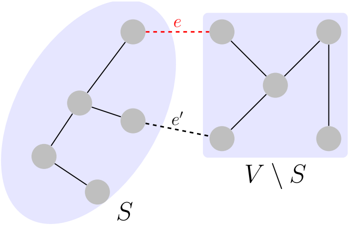


Cut Property (Version II)

$$\text{A cut } (S, V \setminus S)$$

$$\text{Let } e = (u, v) \text{ be } \underline{\text{a}} \text{ lightest edge across } (S, V \setminus S)$$

$$\exists \text{ MST } T \text{ of } G : e \in T$$



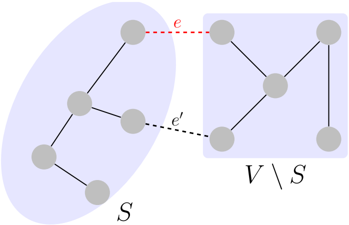
$$T' = \underbrace{T + \{e\}}_{\text{if } e \notin T} - \{e'\}$$

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 $G : e \in T$



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

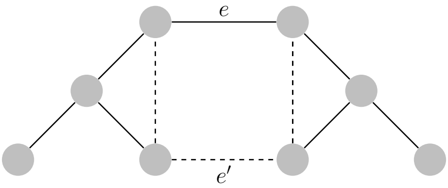
"a" $\to$ "the" $\implies$ "$\exists$" $\to$ "$\forall$"

Cycle Property

Cycle Property

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

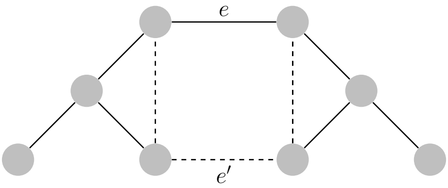
Then $\exists$ MST T of $G : e \notin T$.



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 $G : e \notin T$.

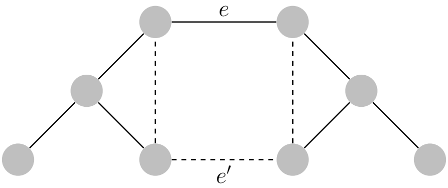


$$T' = \underbrace{T - \{e\}}_{\text{if } e \in T} + \{e'\}$$

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" $\Longrightarrow$ "$\exists$" $\to$ "$\forall$"

Joseph Kruskal (1928 $\sim$ 2010)

Anti-Kruskal Algorithm

Reverse-delete algorithm (wiki; clickable)

Anti-Kruskal Algorithm

Reverse-delete algorithm (wiki; clickable)

Cycle Property

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

Anti-Kruskal Algorithm

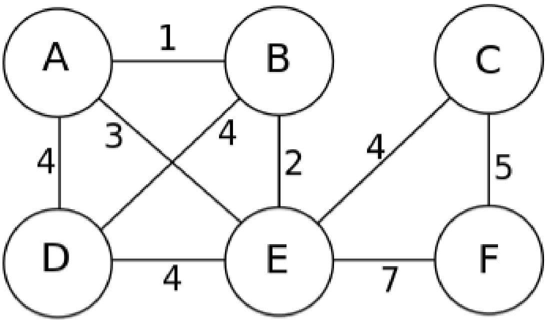
Reverse-delete algorithm (wiki; clickable)

Cycle Property

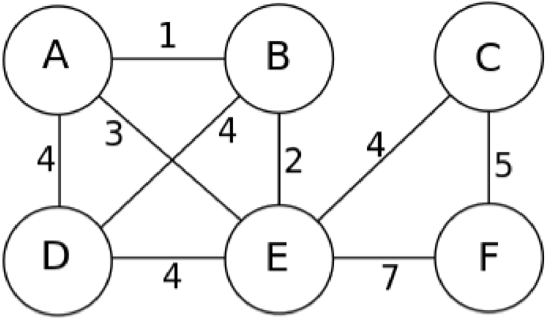
$$T \subseteq F \implies \exists\, T' : T' \subseteq F - \{e\}$$

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

— Kruskal, 1956.


A
B
C
D
E
F
1
4
3
4
2
4
5
4
7

Otakar Borůvka ($1899 \sim 1995$)


A
B
C
D
E
F
1
4
3
4
2
4
5
4
7

Theorem (Uniqueness of MST)

Let G be an edge-weighted undirected graph.
If each edge has a distinct weight, then there is a unique MST of G.

Theorem (Uniqueness of MST)

Let G be an edge-weighted undirected graph.
If each edge has a distinct weight, then there is a unique MST of G.

By Contradiction.

Theorem (Uniqueness of MST)

Let G be an edge-weighted undirected graph.
If each edge has a distinct weight, then there is a unique MST of G.

By Contradiction.

$\exists$ MSTs $T_1 \neq T_2$

Theorem (Uniqueness of MST)

Let G be an edge-weighted undirected graph.
If each edge has a distinct weight, then there is a unique MST of G.

By Contradiction.

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

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

Theorem (Uniqueness of MST)

Let G be an edge-weighted undirected graph.
If each edge has a distinct weight, then there is a unique MST of G.

By Contradiction.

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

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

$$e = \min \Delta E$$

Theorem (Uniqueness of MST)

Let G be an edge-weighted undirected graph.
If each edge has a distinct weight, then there is a unique MST of G.

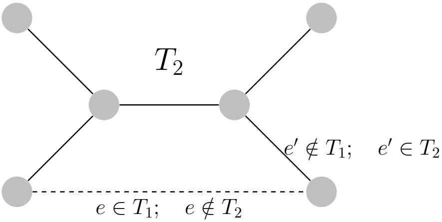
By Contradiction.

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

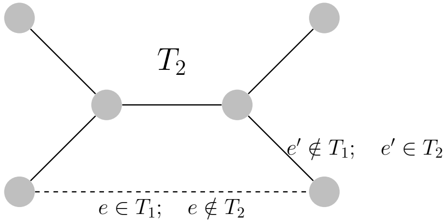
$$\Delta E = \{e \mid e \in T_1 \setminus T_2 \lor e \in T_2 \setminus T_1\}$$

$$e = \min \Delta E$$

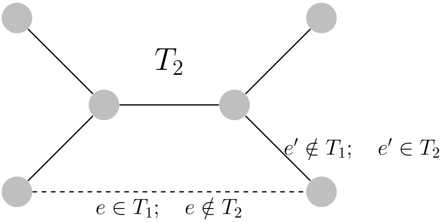
$$e \in T_1 \setminus T_2 \ \textit{(w.l.o.g)}$$



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

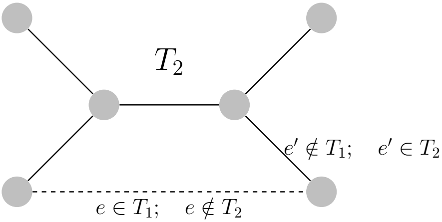


$$T_2 + \{e\} \implies C$$



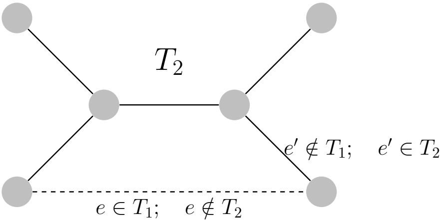
$$T_2 + \{e\} \implies C$$

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



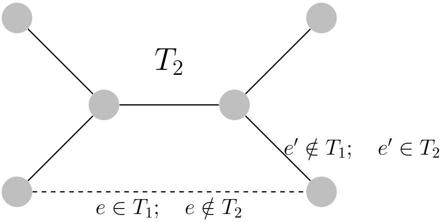
$$T_2 + \{e\} \implies C$$

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



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

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



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

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

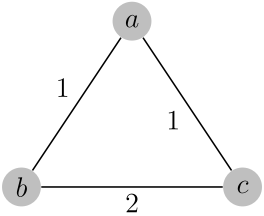
$$T' = T_2 + \{e\} - \{e'\} \implies w(T') < w(T_2)$$

Condition for Uniqueness of MST

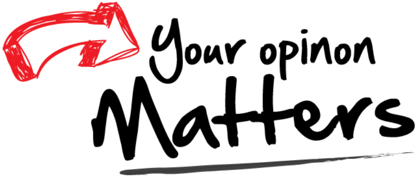
$$\text{Unique MST} \not\Longrightarrow \text{Distinct weights}$$

Condition for Uniqueness of MST

$$\text{Unique MST} \;\not\Longrightarrow\; \text{Distinct weights}$$




Thank
You!



Office 302

Mailbox: H016

hfwei@nju.edu.cn