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)

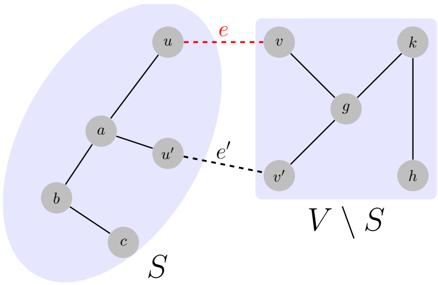
- X is some part of an MST T of G
- Any *cut* $(S, V \setminus S)$ *s.t.* X does *not* cross $(S, V \setminus S)$ Âŋ
- Let e be *a* lightest edge across $(S, V \setminus S)$

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

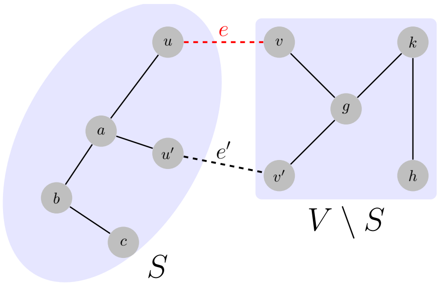
Correctness of Prim's and Kruskal's algorithms.

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 $(S, V \setminus S)$

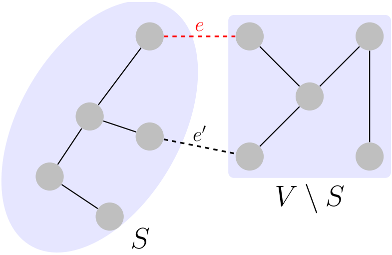
Then e must be in *some* MST of G.

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.

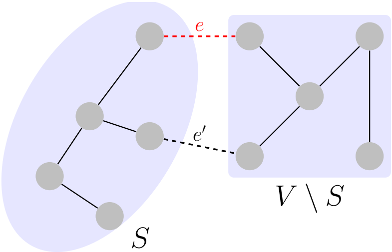


Cut Property (Weak)

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" $\rightarrow$ "the" $\Longrightarrow$ "some" $\rightarrow$ "any"

Converse of Cut Property (Weak)

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

$$\Longrightarrow$$

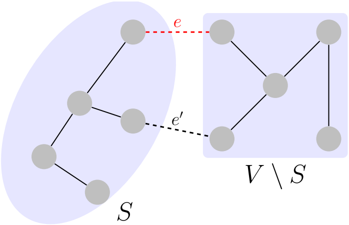
$$e \text{ is a lightest edge across some cut } (S, V \setminus S)$$

Converse of Cut Property (Weak)

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

$$\Longrightarrow$$

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



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

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

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

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)]

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

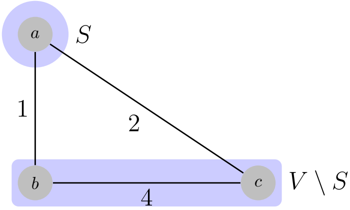
$$(V_1, V_2) : ||V_1| - |V_2|| \leq 1$$

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

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

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

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



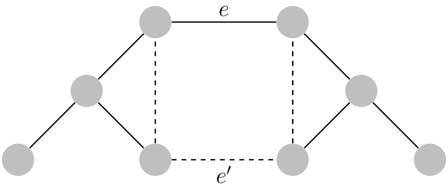
Cycle Property

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

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

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

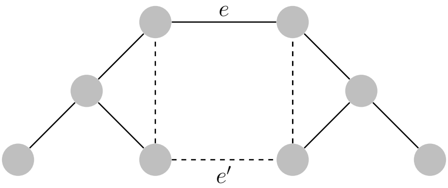
Then $\exists$ MST T of $G : e \notin T$.



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. $\square$

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 unique maximum-weighted edge

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

$e \notin$ 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 is the unique maximum-weighted edge of G

$$\Longrightarrow$$

$e \notin$ any MST of G

By contradiction.

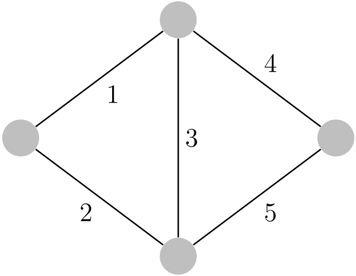
$$T' = T - \{e\} + \{e'\}$$

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)]

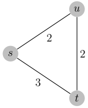
$C \subseteq G, e \in C$, e is the unique lightest edge of $C \implies {}^{?} e \in \forall$ MST



Properties of MST

✓or ✗[Problem: 3.6.15]

1. ✗The shortest-path tree computed by Dijkstra's algorithm is necessarily an MST.
2. ✗The shortest path between two nodes is necessarily part of some MST.



3. ✓Prim's algorithm works correctly when there are negative edges.
4. ✓If e belongs to some MST, then e is a minimum edge across some cut.
5. ✓$w > 0$; Vertex s; shortest-path tree of s and some MST share a common edge [Problem: 6.1.5]
6. ✓$w'(e) = (w(e))^2$ [Problem: 6.2.2]

Uniqueness of MST

Uniqueness of MST [Problem: 3.6.21]

Distinct weights $\Rightarrow$ unique MST.

Solution.

Proof.

By contradiction: two MSTs $T_1 \neq T_2$.

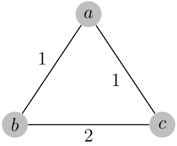
- $\Delta E = \{e \mid e \in T_1 \setminus T_2 \lor e \in T_2 \setminus T_1\}$
- $e = \min \Delta E$. Suppose $e \in T_1 \setminus T_2$
- $T_2 + \{e\} \Rightarrow C$
- $\exists (e' \in C) \notin T_1 \Rightarrow e' \in T_2 \setminus T_1 \Rightarrow e' \in \Delta E \Rightarrow w(e') > w(e)$
- $e' \in T_2 \setminus T_1 \Rightarrow e' \in \Delta E$
- $T' = T_2 + \{e\} - \{e'\} \Rightarrow w(T') < w(T_2)$

□

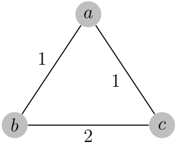
Uniqueness of MST

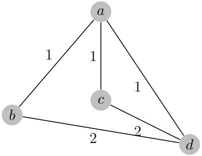
Conditions for Uniqueness of MST [Problem: 3.6.19]

- [Problem: 3.6.19 (a)]: unique MST $\not\Rightarrow$ equal weights



- [Problem: 3.6.19 (c)]: Counterexamples
  - ✗cut: minimum-weight edge across any cut is unique
  - ✗cycle: maximum-weight edge in any cycle is unique





Updating MST

Adding vertex to MST [Problem: 3.6.2]

- $G = (V, E)$; an MST T
- $G' = (V', E')$: $V' = V + \{X\}, E' = E + E_X$; E_X: incident edges to X
- To find an MST T' of G'

Solution.

1. Recomputing $O((m+n)\log n)$
2.
   - There *exists* an MST of G' that includes no edges in $G \setminus T$
   - Run MST alg. on $G'' = (V + \{X\}, T + E_X)$
   - $O(n \log n)$
3. $O(n)$
   - "On Finding and Updating Spanning Tress and Shortest Paths", 1975
   - "Algorithms for Updating Minimum Spanning Trees", 1978

Variants of MST

Feedback edge set: [Problem: 3.6.4]

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

Solution.

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

Variants of MST

Edge weights [Problem: 3.6.15 (8); 3.6.16]

- [Problem: 3.6.15 (8)]: negative edges for Prim algorithm
- [Problem: 3.6.16]: $w'(e) = (w(e))^2$

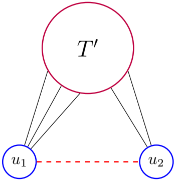
Variants of MST

MST with specified leaves: [Problem: 3.6.7]

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

Solution.

- $G' = G \setminus U$
- MST T' of G'
- attach $\forall u \in U$ to T' (lightest edge)



Variants of MST

ST with specified edges: [Problem: 3.6.10]

- $G = (V, E), S \subset E$ (no cycle in S)
- finding an MST with E as edges

Solution.

- contract each isolated component of S to a *super-vertex*
- $G \to G'$
- find MST of G'

Minimum Spanning Tree (MST)

1. MST vs. Shortest Paths

MST vs. shortest paths

[Problem: 3.6.15]

- [Problem: 3.6.15 (6)]: Dijkstra $\Rightarrow$ SSSP tree $\Rightarrow^{?}$ MST
- [Problem: 3.6.15 (7)]: $s \to t$ shortest path $\Rightarrow^{?} \subseteq \exists$ MST

MST vs. shortest paths

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

Solution

E': lightest edges leaving s

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


THANK YOU