4 Feb 2019

Finishing correctness proof for Min Spanning Tree algorithms.

(1) Grow tree from a root node, always inserting cheapest edge to a new node.

(2) Connect each component to its nearest neighbor; iterate.

(3) Insert edges in order of increasing cost, skipping those that create a cycle.

(A) Kruskal

(B) Prim

(C) Boruvka

Proof technique: Exchange Argument.

Idea: To prove T is optimal, take any other tree T' and devise a way to either (i) improve its cost, or (ii) make it "more similar" to T, while preserving cost. This proves T is optimal because if iterate that operation starting from $T_0 = T'$ we get a sequence

$$T_0,\ T_1,\ T_2,\ \dots,\ T_k = T$$

$$\text{cost}(T') = \text{cost}(T_0) \geq \text{cost}(T_1) \geq \dots \geq \text{cost}(T_k) = \text{cost}(T)$$

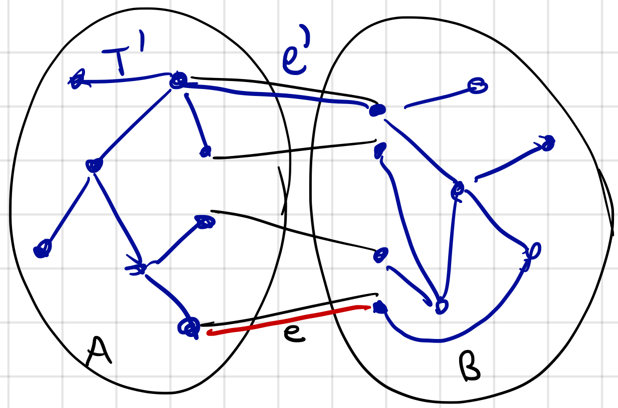
"cut property of MST"

Lemma: If (A,B) is a partition the vertices into two non-empty sets and $e=(u,v)$ is the unique minimum cost edge from A to B then e must belong to every minimum spanning tree of G.

"cut property of MST"

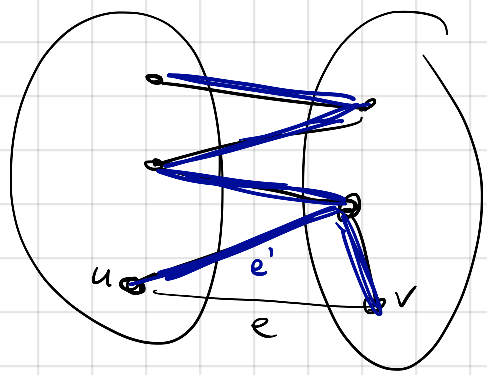
Lemma: If (A,B) is a partition the vertices into two non-empty sets and $e=(u,v)$ is the <u>unique</u> minimum cost edge from A to B then e must belong to <u>every</u> minimum spanning tree of G.

<u>Proof.</u> Suppose $e \notin T'$.



$T' - \{e'\} \cup \{e\}$ is another tree with strictly lower cost.

If T' has more than one edge crossing the cut...



... T' must contain a path from u to v. Walking from u to v along this path, let e' be the first edge that crosses from A to B. Deleting e' from T' separates u, v into separate components. Inserting e reconnects those components.

$\therefore$ $T' - \{e'\} \cup \{e\}$ is a connected subgraph (with $n-1$ edges)

$\therefore$ $T' - \{e'\} \cup \{e\}$ is a spanning tree.

"cut property of MST"

Lemma: If (A,B) is a partition the vertices into two non-empty sets and $e=(u,v)$ is the <u>unique</u> minimum cost edge from A to B then e must belong to <u>every</u> minimum spanning tree of G.

Proving the three algorithms are correct when all edge costs distinct.

(1) Prim: always pick cheapest edge from $\underbrace{\{\text{discovered vertices}\}}_{A}$ to $\underbrace{\{\text{undiscovered vertices}\}}_{B}$

(2) Boruka: in parallel, for every component C_i pick cheapest edge from $\underset{A}{\underset{\shortparallel}{C_i}}$ to $\underset{B}{\underset{\shortparallel}{\underbrace{\text{its complement}}}}$

$\therefore$ every edge added in every Boruvka phase must belong to every MST.

(3) Kruskal: "Like Boruvka in slow motion."
Repeatedly finds cheapest edge whose endpoints belong to distinct forest components, say C_i and C_j.

Cut property with $A=C_i$, $B=V(G)-C_i$ justifies that this edge must belong to every MST.

<u>Remarks.</u> ① If G doesn't have distinct edge costs.

<u>Lemma.</u> If T is a spanning tree of G, and $\forall\, e \in E(T)$, e is one of the min-cost edges between the two conn components of $T-\{e\}$ then T is a minimum spanning tree.

(2) Running times of all these algorithms:

$O(m \log n)$. (See book if curious.)
§§ 4.5 – 4.6

(3) One more useful property:

The Cycle Property. If C is a cycle in G, and e is the unique maximum cost edge in C, then e doesn't belong to any Min Spanning Tree.

(Proof again uses exchange argument.)