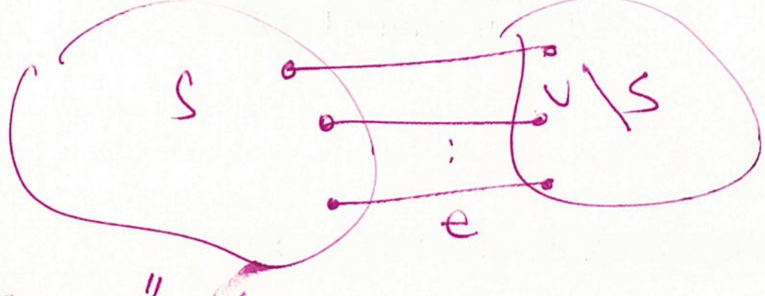
Oct 22

# Cut Property Lemma

ASSUME: All $c_e$'s are distinct

For all cuts $(S, V\setminus S)$ s.t. $S \neq \emptyset$
$V \setminus S \neq \emptyset$

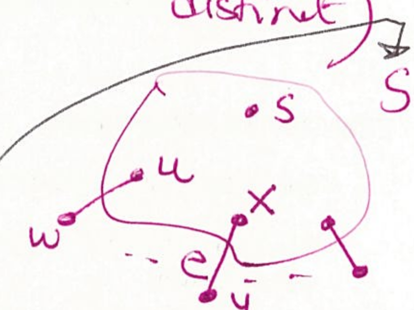



Consider all "crossing" edges
Let e be the cheapest crossing edge
$\Rightarrow$ e is in ALL MSTs of G.

---

Assume Cut Property Lemma is true (+ $c_e$'s are distinct)

THM 1: Prim's algo is correct




Pf (idea) Consider run of Prim's when it is about to add e to T

Goal: Will show that e is the cheapest crossing edge across some cut.

Apply Cut property Lemma to the cut $(S, V\setminus S)$, where S is as in Prim's ~~algo~~ algo

Claim 1: e is the cheapest crossing edge $(S, V\setminus S)$. (from def of Prim's algo)

Claim 2: $S \neq \emptyset$ $(x \in S)$

Claim 3: $S \neq V$ $(y \notin S)$

$\Rightarrow$ every edge added by Prim's is correct/"safe" (as e is in ALL MSTs)

Q: Are we done?

A: Need to argue that $T$ is connected

Claim 4: At the end of each iteration $(S, T)$ is connected

Ex $\Rightarrow$ At ~~at~~ the end, $(V, T)$ is connected

Claims 1+2+3+4 $\Rightarrow$ THM 1 ∎

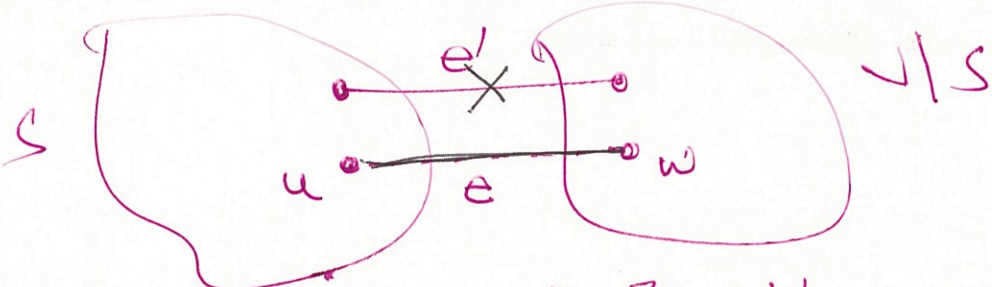
---

Pf (idea) ~~By~~ [Cut Property Lemma]

By contradiction

Assume $\exists$ a cut $(S, V\setminus S)$ & an MST $T$ s.t. (*)

the cheapest crossing edge $e$ is NOT in $T = (V, E')$




→ Since $T$ is connected $\Rightarrow$ $\exists$ exist crossing edge $e' \in E'$

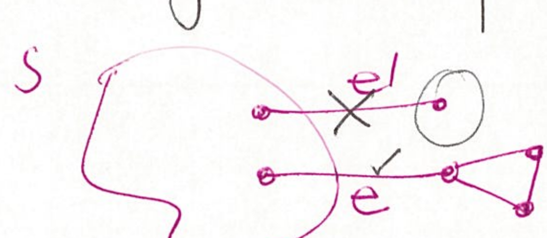
Consider $T' = (V, (E' \setminus \{e'\} \cup \{e\}))$

$c(T') = c(T) - c_{e'} + c_e$

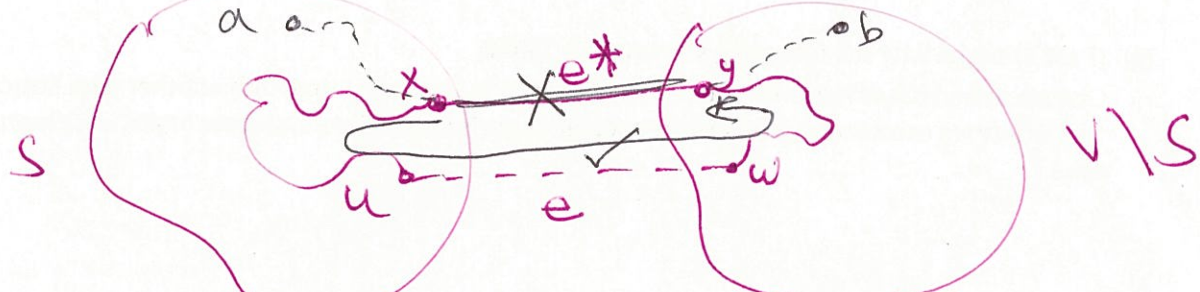
$< c(T)$ $\Rightarrow$ contradicts $T$ is an MST. ∎

Obs: $c_{e'} > c_e$ ($e$ being cheapest crossing edge + distinct edge costs)

Q: What is wrong in the "proof" above??



Fix! Pick e' more carefully




Since T is connected $\Rightarrow \exists$ u–w path in T

As $u \in S$ & $w \notin S$ $\Rightarrow \exists$ a crossing edge $e^* = (x,y)$

Define $T' = (V, (E' \setminus \{e^*\}) \cup \{e\})$

Claim 1: $c(T') = c(T) - c_{e^*} + c_e < c(T)$ ∎

Claim 2: T' is connected $\rightarrow$ every $a, b \in V$ are connected in T'

Case 1: a–b path in T doesn't use $e^*$ ✓

Case 2: a–b path in T uses $e^*$ $\Rightarrow$ take "longer" route ✓ ∎

---

THM 2: Kruskal's algo is correct.

~~Pf~~ Consider the algo when it is about to add an edge $e = (u,w)$

Goal: e is the cheapest crossing edge for some cut $(S, V\setminus S)$

Def: S be the set of vertices connected to u using only the edges in T.