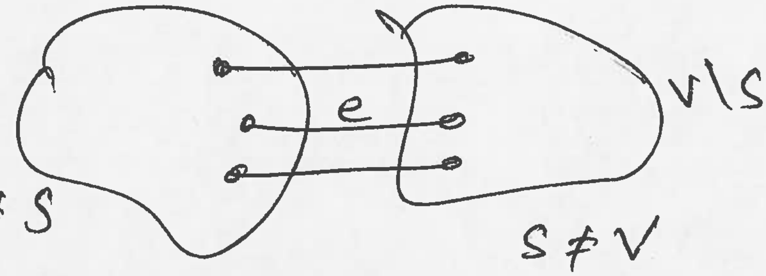
Oct 31

CUT PROPERTY LEMMA

Assume all c_e are distinct (ie $c_e \neq c_{e'}$ $\forall e \neq e' \in E$)

Then ~~fact~~ for all cuts $(S, V \setminus S)$



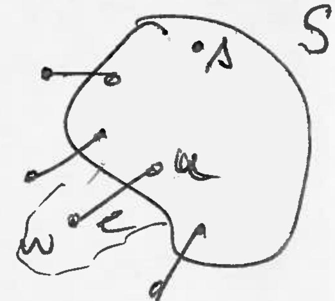
Let e be the cheapest crossing edge
$\Rightarrow$ e is in <u>ALL</u> MSTs for G.

<u>THM:</u> Prim's algo is correct

<u>Pf</u> (idea) Consider the point where algo is about to add e to T

<u>Goal:</u> Argue that adding e is "safe"

Recall



Add cheapest crossing edge $e = (u, w)$ to T

Pick the cut $(S, V \setminus S)$ where S is as defined in algo.

By algo definition, e is the cheapest crossing edge for $(S, V \setminus S)$

<u>Claim 1:</u> $S \neq \{\}$ (as $u \in S$) <u>Claim 2:</u> $S \neq V$ (as $w \notin S$)

$\Longrightarrow$ adding e is safe.

Cut property lemma

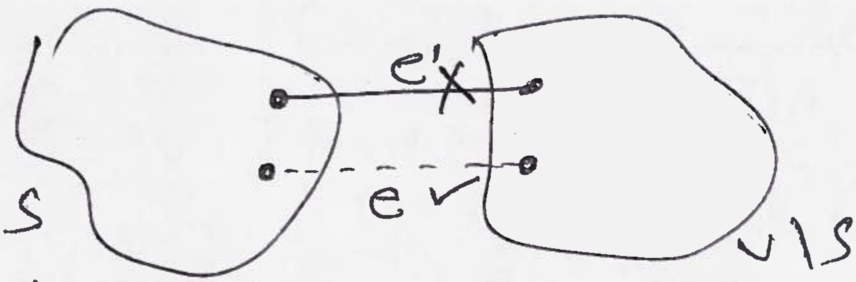
$\rightarrow$ Need to argue: The final T is s.t. (V, T) is a tree.

<u>Claim 3:</u> At end of every iteration (S, T) is a tree ⊠
(EX)

Pf of Cut Property Lemma: For contradiction assume

$\exists \{\} \neq S \neq V$ & $\exists$ an MST T s.t. the cheapest crossing edge e for $(S, V \setminus S)$ is NOT in T.

Will argue: exists another spanning tree T' s.t. $c(T') < c(T)$



As T is a spanning tree $\Rightarrow \exists$ a crossing edge e' s.t. $e' \in T$

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

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

$< c(T)$

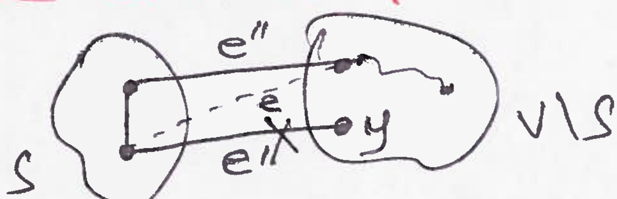
but $c_e < c_{e'}$ (as all c_e's are distinct & e is the cheapest crossing edge)

$\Rightarrow$ done

NOT correct!

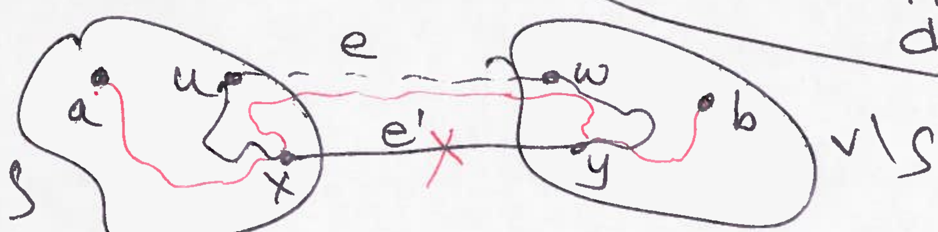
Problem: T' ~~is~~ need not be connected!

In the case drop e''.



Dropping e' would ~~discon~~ disconnect y!

fixed (pf) $e = (u, w) \notin T$



As T is a spanning tree $\exists$ a u-w path in T $\Rightarrow$ $\exists$ crossing edge $e' = (x, y)$ in this path

Pick this e' to ~~drop~~ drop

$T' = T \setminus \{e'\} \cup \{e\} \Longrightarrow c(T') < c(T)$ as before

Claim: T' is still connected.

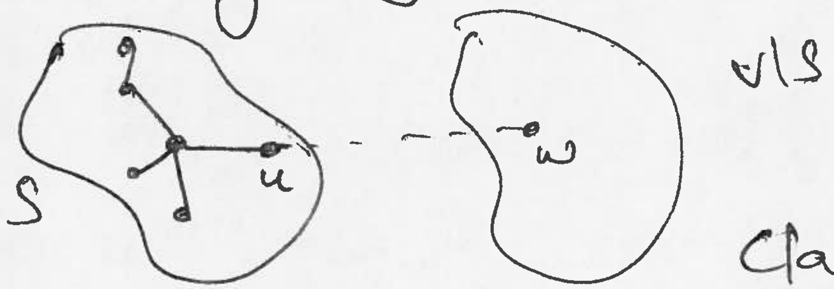
Case 1: If a-b path in T does not use e' $\Rightarrow$ ✓

Case 2: ________ does use e'
$\Rightarrow$ take longer path in T'. ∎

THM: Kruskal's algo is correct [Recall: consider all edges in increasing order of cost & add it to T if doing so doesn't add a cycle)

Pf: Consider the time when the algo is about to add $e = (u,w)$ to T

Define S to be set of vertices connected to u using only edges in T.



Claim 1: $S \neq \{\}$
($\because u \in S$)

Claim 2: $S \neq V$
($\Leftarrow w \notin S$)

Claim 3: e is the cheapest crossing edge.
($\Leftarrow$ e is the first crossing edge for $(S, V\setminus S)$ considered by Kruskal's algo)

Claims 1+2+3 $\Rightarrow$ choice of e is safe.