Algorithms: Design and Analysis, Part II

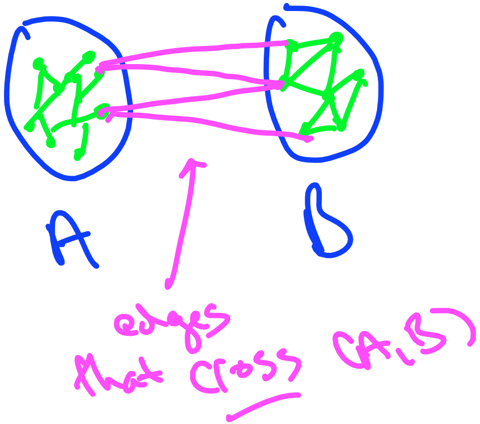
Minimum Spanning Trees

Correctness of Prim's Algorithm (Part I)

Cuts

<u>Claim</u>: Prim's algorithm outputs a spanning tree.

<u>Definition</u>: a <u>cut</u> of a graph $G=(V,E)$ is a partition of V into 2 non-empty sets.



Quiz on Cuts

Question: roughly how many cuts does a graph with n vertices have?

(A) n

(B) n^2

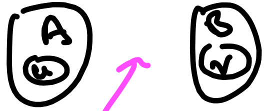
(C) 2^n (for each vertex, choose whether in A or in B)

(D) n^n

Empty Cut Lemma

Empty Cut Lemma: a graph is <u>not</u> connected $\iff$ $\exists$ cut (A,B) with <u>no</u> crossing edges.

Proof: ($\Leftarrow$) Assume the RHS. Pick any $u \in A$ and $v \in B$.



no crossing edges so no u-v path

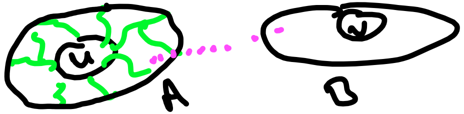
Since no edges cross (A,B), there is no u-v path in G. $\Rightarrow$ G not connected

($\Rightarrow$) Assume the LHS. Suppose G has no u-v path.

Define A = {vertices reachable from u in G} (i.e., u's connected component)

B = {all other vertices} (i.e., all other connected components)

<u>Note</u>: no edges cross the cut (A,B) (otherwise A would be bigger!)

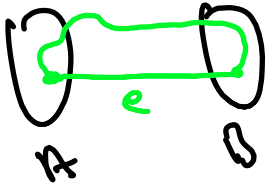


QED!

Tim Roughgarden

Two Easy Facts

Double-Crossing Lemma: Suppose the cycle $C \subseteq E$ has an edge crossing the cut (A,B): then so does some other edge of C.



Lonely Cut Corollary: if e is the only edge crossing some cut (A,B), then it is not in any cycle.

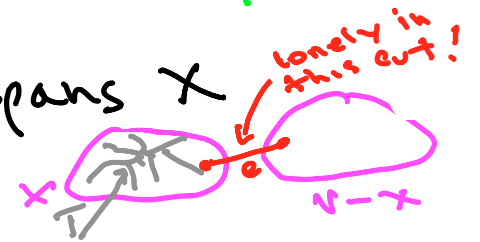
[if it were in a cycle, some other edge would have to cross the cut!]

Proof of Part I

[not claiming MST yet]

Claim: Prim's algorithm outputs a spanning tree.

Proof: ① algorithm maintains invariant that T spans X
[straightforward induction — you check]



② can't get stuck with $X \neq V$ (otherwise the cut $(X, V-X)$ must be empty; by Empty Cut Lemma input graph G is disconnected)

③ no cycles ever get created in T. Why? Consider any iteration, with current sets X and T. Suppose e gets added.

Key point: e is the first edge crossing $(X, V-X)$ that gets added to T $\Rightarrow$ its addition cannot create a cycle in T [by Lonely Cut Corollary]

QED!