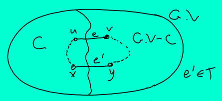
Given graph G with edge weights, let $A \subseteq G.E$ be a subset of some MST of G (proper). Let C be any connected component of $(G.V, A)$, let e be any minweight edge with one endpoint in C and the other not in C.

Show that e is <u>safe</u>, that is $A \cup \{e\}$ is a subset of an MST.

<u>Proof</u>: Let $T \supseteq A$ be an MST of G. If $e \in T$ then done, so assume otherwise.



$T \cup \{e\}$ has a cycle containing e.

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

$wt(\overbrace{(T \cup \{e\}) - \{e'\}}^{T'})$

$wt(T') = wt(T) + \underbrace{wt(e) - wt(e')}_{\leq 0}$

$wt(T') \geq wt(T)$

$\therefore wt(e) - wt(e') \geq 0$

$\therefore wt(e) \geq wt(e')$

$wt(e) \leq wt(e')$ by assumption

$\therefore wt(e) = wt(e')$

$\therefore wt(T') = wt(T)$

$\therefore T'$ is an MST (containing e)

$\therefore e$ is safe //

Shortest path (single source)

Input $(G, w, s \in G.V)$

G is a digraph

$w : G.E \to \mathbb{R}$

such that $w(e) \geq 0 \quad \forall e \in G.E$

<u>Output</u>: attributes

$v.d$ — shortest $s \rightsquigarrow v$ distance

$v.\pi$ — predecessor to v along a shortest path.

Key idea: if $v_i \rightsquigarrow v_j$ is a shortest path going through some v_k:

$v_i \rightsquigarrow v_k \rightsquigarrow v_j$

then these are shortest paths.

Dijkstra's Algo (G, w, s)

Initially

$s.d := 0$

$s.\pi := Nil$

$\left.\begin{array}{l} v.d := \infty \\ v.\pi := Nil \end{array}\right\}$ all $v \neq s$

Insert all vertices into a min-priority queue Q with d-attr as key.

while Q not empty

$u :=$ DeleteMin(Q)

for every edge (u, v) s.t. $v \in Q$

if $v.d > \boxed{u.d +} w(u,v)$:

DecreaseKey $\longrightarrow$ $v.d := u.d + w(u,v)$

$v.\pi := u$

only difference from Prim's algo

worst-case

<u>Time</u>: n inserts $O(n)$

m DecreaseKey's $O(m \lg n)$

n DeleteMin's $O(n \lg n)$

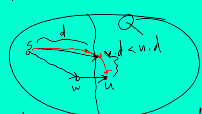
Bin heap: $O((n+m)\lg n)$

<u>Fib heap</u>: $O(m + n \lg n)$

Correctness sketch:

<u>Claim</u>: when u is removed from Q, its d-value is correct.

suppose otherwise



$v.d < u.d$, so u not removed from Q yet ⚡.