L7 3.4.17

Topic: Dynamic Graphs I

- Operations: Insert/Delete Edges
  Connectivity(u,v)

- Flavour: Fully Dynamic
  Partially: - Incremental
  - Decremental

Trees: • $O(\log n)$ worst case / op fully dynamic.
• $O(1)$ amortized decremental (Eppstein et al. 96) old, at least not young

Plane graphs: $O(\log n)$

General Graph:

- Hope $O(\log^{O(1)} n)$/op
- $O(\log^2 n)$ update
  $O(\log n / \log\log n)$ query
- $O(\sqrt{n})$ update
  $O(1)$ query
  } Eppstein et. al. 97
- $O(n^{0.49})$ update Wulf-Nielsen 2017

Decremental: $O(m \log n + \text{polylog}(n) n + \#\text{queries})$

Incremental: Union-Find D.S.

$O(\alpha(m,n))$/op ~1975

α: Inverse A.

Lowerbound: Update or Query needs $\Omega(\log n)$

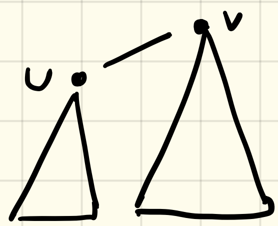
Dynamic Connectivity on Trees

Approach: Link-Cut-Trees

Euler Tour Trees

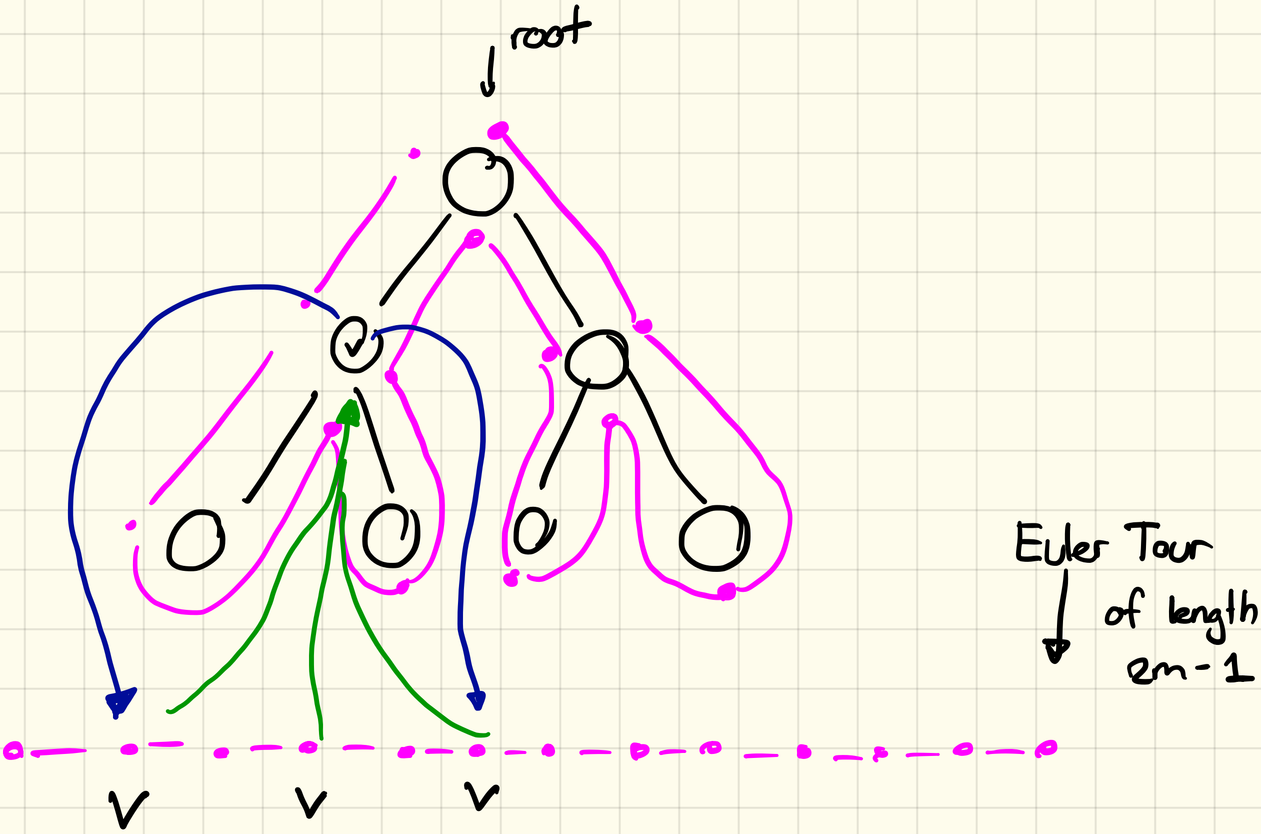
Euler Tour Tree:

- MAKE_TREE(v): New isolated tree • v
- Link(u,v):



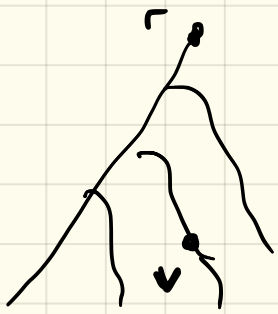
- Cut(v): Delete edge (v, parent(v))
- Find_Root(v)

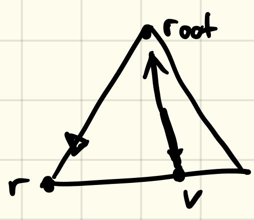
Idea: maintain Euler tour around the tree



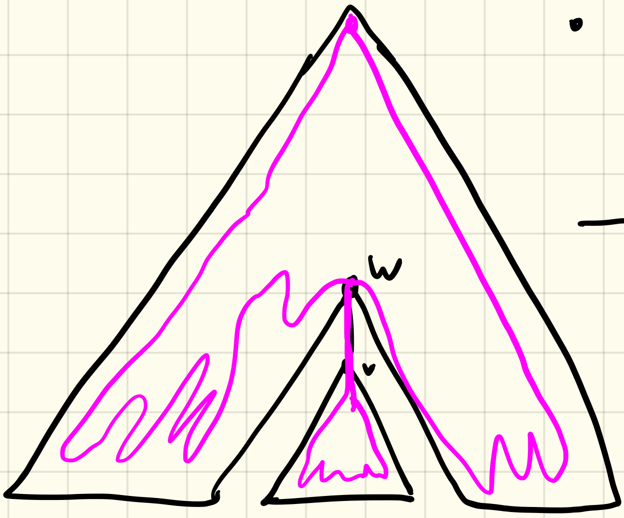
→ Build Balanced Binary Search Tree on Euler Tour ordered by order in tour.

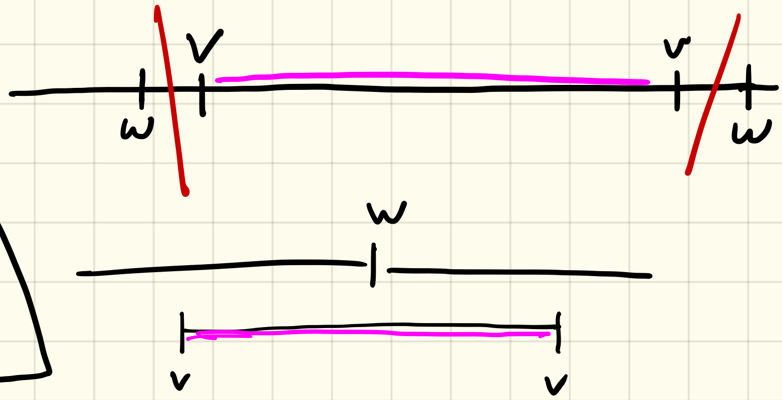
FindRoot(v):
- Start in first visit to v on tour
- Walk up to root of BST
  Walk Left to r



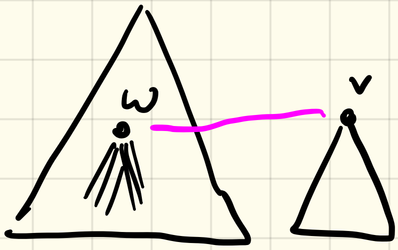


- Make_Tree(v) : 'trivial'
- Cut(v):
  - Split of BST in $O(\log n)$
  - Remove one w in $O(\log n)$





- Link(v,w):
  - Add one w
  - Merge BST



Note: If v is not a root, one can make it root!

- Connectivity(v,w) =

$$\text{Find\_Root}(v) \overset{?}{=} \text{Find\_Root}(w)$$

Possible Augmentation of nodes
(min, max, sum) between first & last visit.

Fully Dynamic Graphs   Holm et. al

$\mathcal{O}(\log^2 n)$ amortized time

Idea: Maintain Spanning Forest

Problem Deletion of Edge in SPT

→ Hierarchically divide connected components of G

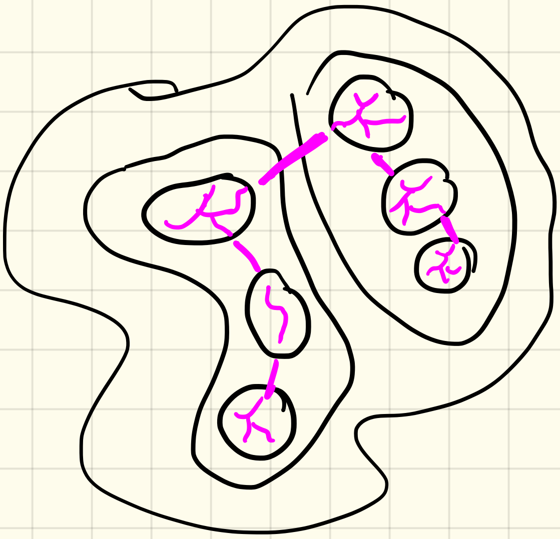
- $\log n$ levels of edges partition of E

G_i = subgraph of G of edges level $\leq i$

$G_0 \subseteq G_1 \subseteq \ldots G_{\log n} = G$

Invariant 1 Every connected component of G_i has size $\leq 2^i$ (vertices)

F_i = spanning forest of G_i



Invariant 2

$F_i \stackrel{!}{=} F_{\log n} \cap G_i$

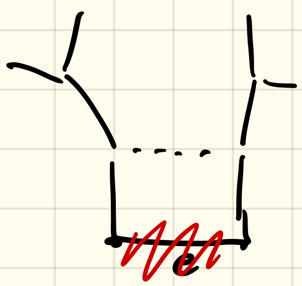
Query: Euler tour tree query on $F_{\log n}$

Insert$(e=(v,w))$:

- Add e to incidence lists of v,w
- e level $= \log n$
- If v,w were disconnected, add e to $F_{\log n}$ (reroot to v)

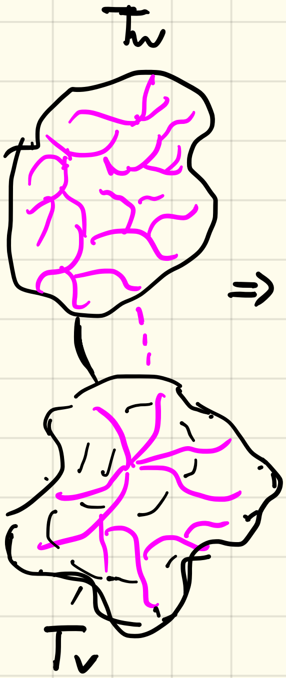
Delete$(e=(v,w))$

- Remove e from incidence lists
- If $e \notin F_{\log n} \rightarrow$ Done
- If $e \in F_{e\text{-level}} \rightarrow e$-level
  - delete e from $F_{e\text{-level}}, \ldots, F_{\log n}$
  - if there is a replacement edge, at level x it is good on $F_x, \ldots\ldots, F_{\log n}$



For $i = e\text{-level}, \dots, \log n$

T_v, T_w be the trees of F_i containing v&w



w.l.o.g. $\boxed{|T_v| \le |T_w|}$ here we need size info of ETT

$|T_v| + |T_w| \le 2^i \Rightarrow |T_v| \le 2^{i-1}$?

$\Rightarrow$ Can "afford" to push T_v into $i-1$ level.

For each edge $e' = (x,y)$ with $x \in T_v$ and level i

IF $y \in T_w$: add e' to $F_i, \dots, F_{\log n}$

and <u>STOP</u>

ELSE : e'- level $= i-1$ since $y \in T_v$

add e' to F_{i-1}