

TODAY: Dynamic graphs II

- fully vs. partially dynamic
- Euler-tour trees
- $O(1)$ decremental connectivity in trees
- $O(\lg^2 n)$ fully dynamic connectivity
- survey

Dynamic connectivity:

maintain undirected graph subject to

- insert/delete edges or vertices (with no edges)
- connectivity(v, w): is there a $v \rightarrow w$ path?
or $()$: is the graph connected? (\approx same)

Dynamic graph problems: characterized by updates

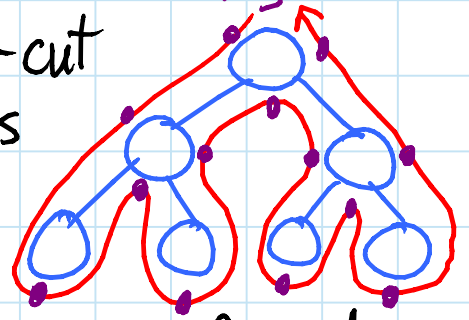
- fully dynamic: insert & delete \leftarrow default
- partially dynamic:
 - incremental: just insert
 - decremental: just delete

Dynamic connectivity results:

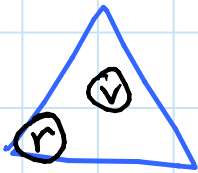
- trees: $O(\lg n)$ → link-cut [L19] & Euler tour trees
- decremental: $O(1)$ amortized → **TODAY**
- plane graphs: $O(\lg n)$
↳ embedded planar [Eppstein, Galil, Italiano, Spencer - JCSS 1996]
- general graphs, amortized:
 - **OPEN**: $O(\lg n)$ update & query
 - $O(\lg^2 n)$ update, $O(\frac{\lg n}{\lg \lg n})$ query → **TODAY**
[Holm, de Lichtenberg, Thorup - J.ACM 2001]
 - $O(\lg n (\lg \lg n)^3)$ update, $O(\frac{\lg n}{\lg \lg \lg n})$ query
[Thorup - STOC 2000]
 - incremental: $O(\alpha(m, n))$ via union-find [Tarjan - JACM 1975]
 - decremental: $O(m \lg n + n \text{ poly} \lg n + \# \text{queries})$ total
[Thorup - J.Alg. 1999]
- worst case: (general graphs)
 - $O(\sqrt{n \frac{(\lg \lg n)^2}{\lg n}})$ update, $O(1)$ query
[Kejlberg-Rasmussen, Kopelowitz, Pettie, Thorup - ESA 2016]
 - $O(\lg^4 n)$ update, $O(\frac{\lg n}{\lg \lg n})$ query correct w.h.p. [Gibb, Kapran, ...]
 - incremental: $\Theta(x)$ updates \sim King, Thorn - arXiv 2015
 $\Rightarrow \Theta(\frac{\lg n}{\lg x})$ queries [Alstrup, Ben-Amran, Rauhe - STOC 1999]
- lower bounds: $\Omega(\lg n)$ update or query } even for paths!
 - $O(x \lg n)$ update $\Rightarrow \Omega(\frac{\lg n}{\lg x})$ query
 - $O(x \lg n)$ query $\Rightarrow \Omega(\frac{\lg n}{\lg x})$ update[L21] & [Patrascu & Demaine - STOC 2004/SICOMP 2006]
- points on trade-off curve
- **OPEN**: $o(\lg n)$ update & $\text{poly} \lg n$ query?

Euler-tour trees: [Henzinger & King - STOC 1995]

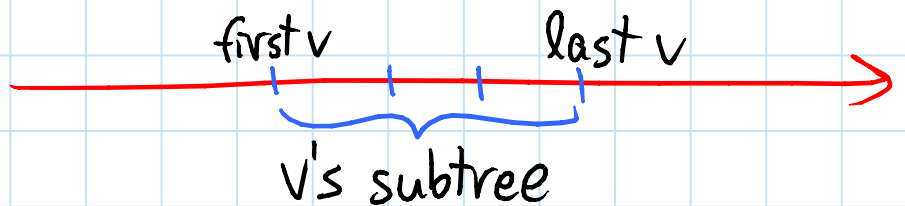
- simpler dynamic trees than link-cut
- aggregates over subtrees, vs. paths
- Euler tour [L15] around tree
 - visits each edge twice
- store node visits by Euler tour in balanced BST
- each node stores pointers to first & last visits



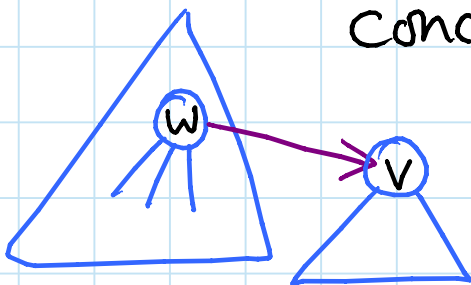
- findroot(v): start at first visit to v in BST
walk up to root of BST
walk left to min of BST
→ first visit of root node



- cut(v): split BST at v's first & last visits
concatenate "before" & "after" trees



- link(v, w): split w's BST before w's last visit
concatenate "before last w",
new single (w),
v's BST, and
"after last w"

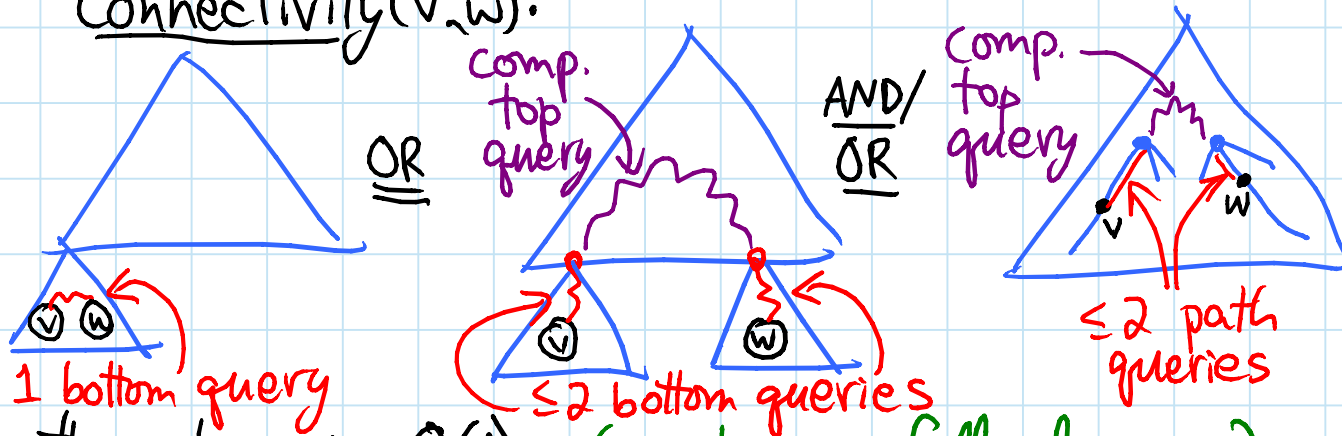


- connectivity(v, w): $\text{findroot}(v) \stackrel{?}{=} \text{findroot}(w)$
- subtree aggregate(v): (min/max/sum/etc.)
range query in BST between first & last visit
- $O(\lg n)$ time/op.

Decremental connectivity in a tree: ^{rooted} [Alstrup, Secher, Spork - IPL 1997]

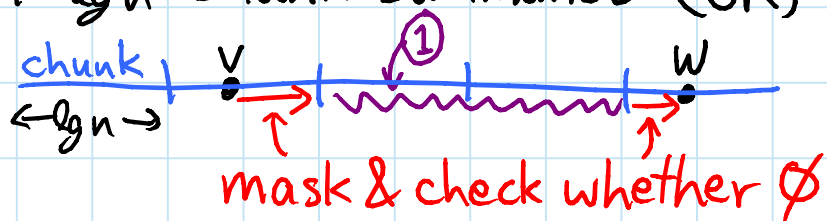
- $O(1)$ amortized, assuming all $n-1$ edges deleted
- ① $O(\lg n)$ via link-cut or Euler-tour trees
(simpler way: maintain explicit node \rightarrow component id. & relabel the smaller side)

- ② leaf trimming: cut below maximally deep nodes with $> \lg n$ descendants
 \Rightarrow top tree has $O(\frac{n}{\lg n})$ leaves/branching nodes
 - use ① on compressed top tree
 - connectivity(v, w):



- ③ bottom tree in $O(1)$: (worst case, fully dynamic)
 - store bit vector of which edges don't exist
 - preprocess mask of each node's ancestors (1 word)
 - XOR masks for v & w , mask, check whether \emptyset

- ④ path in $O(1)$ amortized:
 - split path into $\frac{n}{\lg n}$ chunks of length $\lg n$
 - store each chunk as bit vector (1 word)
 - use ① on $\frac{n}{\lg n}$ chunk summaries (OR)
 - query:



$O(\lg^2 n)$ dynamic connectivity: [Holm et al. - J.ACM 2001]

- idea:
 - store spanning forest with Euler-tour trees
 - hierarchically divide connected components
- $O(\lg n)$ levels of spanning forests

→ charging mechanism

- level of edge starts at $\lg n$, only decreases → \emptyset
- G_i = subgraph of edges at level $\leq i$
- ⇒ $G_{\lg n} = G$
- INVARIANT 1: every conn. comp. of G_i has $\leq 2^i$ vxs.
- F_i = spanning forest of G_i
 - store using Euler-tour tree DS
- ⇒ $F_{\lg n}$ is desired spanning forest of G
- INVARIANT 2: $F_\emptyset \subseteq F_1 \subseteq \dots \subseteq F_{\lg n}$ i.e. $F_i = F_{\lg n} \cap G_i$
i.e. $F_{\lg n}$ is a min. spanning forest w.r.t. level

insert($e = (v, w)$): $O(\lg n)$

- add e to v & w incidence lists
- $e.$ level = $\lg n$
- if v & w disconnected in $F_{\lg n}$:
 - add e to $F_{\lg n}$ (link)

(reroot to make v root via cyclic shift)

connectivity: $O\left(\frac{\lg n}{\lg \lg n}\right)$

- make $F_{\lg n}$ B-trees with branching factor $\Theta(\lg n)$
- ⇒ $O\left(\frac{\lg n}{\lg \lg n}\right)$ findroot & $O\left(\frac{\lg^2 n}{\lg \lg n}\right)$ update
depth depth · branching factor

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

- remove e from v & w incidence lists
 - if e is in $F_{\lg n}$: (v .parent = w or vice versa)
 - $\lg^2 n \rightarrow$ - delete e from $F_{e.level}, \dots, F_{\lg n}$ (cut)
 - look for replacement edge to reconnect v & w
 - can't be any edges with level $< e.level$
 - \Rightarrow find min. possible level $\geq e.level$ [Invariant 2]
 - for $i = e.level, \dots, \lg n$:
 - let T_v & T_w be trees of F_i containing v & w resp.
 - relabel so that $|T_v| \leq |T_w|$
 - Invariant 1 (before) $\Rightarrow |T_v| + |T_w| \leq 2^i \Rightarrow |T_v| \leq 2^{i-1}$
 - \Rightarrow can "afford" to push all of T_v down to level $i-1$
 - for each level- i edge $e'=(x,y)$ with $x \in T_v$:
 - if $y \in T_w$: add e' to $F_i, F_{i+1}, \dots, F_{\lg n}$
return (found replacement)
 - else ($y \notin T_w$): $e'.level = i-1$ \leftarrow charge
 - Euler-tour tree augmentation: \uparrow & add e' to F_{i-1}
 - \rightarrow - subtree sizes to test $|T_v|$ vs. $|T_w|$ in $O(1)$
 - \rightarrow - for each node v in tree of F_i :
 - does v 's subtree contain any nodes incident to level- i edges?
 - \Rightarrow can find next level- i edge incident to $x \in T_v$ in $O(\lg n)$ time (successor, skipping over empty subtrees)
- \Rightarrow time: $O(\lg^2 n + \# \text{charges} \cdot \lg n)$
- each inserted edge charged $\leq \lg n$ times

k-connectivity: vertex or edge

- disjoint paths between pairs of vertices:
 - 2-edge: $O(\lg^4 n)$ - 2-vertex: $O(\lg^5 n)$ [Holm et al. - JACM 2001]
 - **OPEN**: $\text{poly} \lg n$ for $k=O(1)$? $k=\text{poly} \lg n$?
 - planar decremental: $O(\lg^2 n)$ 3-edge-conn.
[Giannaresi & Italiano - Algorithmica 1996]
- worst case: [Eppstein et al. - J.ACM 1997]
 - 2-edge-conn.: $O(\sqrt{n})$ - 2-vertex-conn.: $O(n)$
 - 3-edge-conn.: $O(n^{2/3})$ - 3-vertex-conn.: $O(n)$
 - $k=4$: $O(n \cdot \alpha(n))$
 - $O(1)$ -edge-conn.: $O(n \lg n)$
- whole graph \approx min cut = max flow
 - $O(\text{poly} \lg n)$ -edge-conn. (& min cut up to that size):
 $O(\sqrt{n} \cdot \text{poly} \lg n)$ [Thorup - STOC 2001]
 - **OPEN**: $\text{poly} \lg n$ for $k=O(1)$? $k=\text{poly} \lg n$?

Minimum spanning forest: (MST on each conn.comp. as dyn. tree)

- $O(\lg^4 n)$ update [Holm, de Lichtenberg, Thorup - J.ACM 2001]
- worst case: $O(\sqrt{n})$ update [Eppstein et al. - J.ACM 1997]
- plane graphs: $O(\lg n)$ [Eppstein et al. - J.ACM 1992]
- can use to solve bipartiteness: is graph 2-colorable?

Planarity testing: insert e or report planarity violation

- $O(n^{1/2})$ [Eppstein, Galil, Italiano, Spencer - JCSS 1996]
- planar: $O(\lg^2 n)$ [ditto]
- incremental: $O(\alpha(m, n))$ amortized [la Poutre - STOC 1994]

Directed graphs:

Transitive closure: is there a $v \rightarrow w$ directed path?

- bulk update: insert/delete vertex & incident edges
- $O(n^2)$ am. bulk update, $O(1)$ worst-case query

[Demetrescu & Italiano - Focs 2000; Roditty - SODA 2003]

- same, worst case [Sankowski - Focs 2004]
- optimal if explicitly storing trans. closure matrix
- **OPEN**: $o(n^2)$ worst-case update?

- $O(m\sqrt{n} \cdot t)$ am. bulk update, $O(\sqrt{n}/t)$ w.c. query for any $t = O(\sqrt{n})$ [Roditty & Zwick - Focs 2002]

- $O(m + n \lg n)$ am. bulk update, $O(n)$ w.c. query [Roditty & Zwick - STOC 2004]

- **OPEN**: full trade-off: update · query = $O(mn)$ or $O(n^3)$

- acyclic: $O(n^{1.575} \cdot t)$ update, $O(n^{0.575}/t)$ query, $t = O(\sqrt{n})$

- decremental: $O(n)$ am. update, $O(1)$ w.c. query [Demetrescu & Italiano - Focs 2000]

All-pairs shortest paths: weight of shortest $v \rightarrow w$ path

- $O(n^2 (\lg n + \lg^2(1 + \frac{m}{n})))$ am. bulk update, $O(1)$ w.c. query

[Thorup - SWAT 2004] improving [Demetrescu & Italiano - STOC 2003]

- **OPEN**: $O(n^2)$ or $o(n^2)$ update, even undirected graphs?

- $O(n^{2.75})$ w.c. update, $O(1)$ query [Thorup - STOC 2005]

- unweighted: $O(m\sqrt{n} \cdot \text{poly} \lg n)$ am. update.

$O(n^{3/4})$ w.c. query [Roditty & Zwick - ESA 2004]

- undirected, unweighted & $(1+\epsilon)$ -approx.: [Roditty & Zwick

$O(\sqrt{m}n \cdot t)$ am. update, $O(\sqrt{m}/t)$ w.c. query, $t = O(\sqrt{n})$ - Focs 2004]

- static & $(1+\epsilon)$ -approx.: distance oracles [...]