Lecture 5

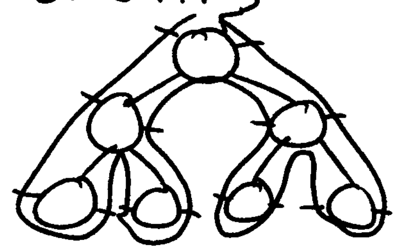
Augmenting link-cut trees

- can maintain aggregates on paths of tree
  Σ, min, max, etc.
- key: link-cut trees store paths

$\Rightarrow$ e.g. weighted shortest-path weight to root

- min important for fast network flow algs.

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



- simple dynamic trees DS
- aggregates on subtrees
- Euler tour = walk around tree
  - visit each edge exactly twice
- store Euler-tour node visitations in balanced BST
- each node stores ptrs. to first & last visits in Euler tour
- findroot(v): min node in balanced BST
- cut(v): split BST at v's first & last visits
  concatenate "before v" & "after v" trees
- link(v, w): split w's BST before last visit to w
  concatenate "before last w" tree, new single w, v's BST, & "after last w" tree
- $O(\lg n)$/operation

Dynamic connectivity: maintain undirected graph
- insert/delete edges/vertices (with no edges)
- connectivity(v, w): is there a path $v \to w$?
  or (G): connected graph? ($\approx$ same)

Known bounds:
- $O(\lg n)$ for trees [link-cut; Euler tour]
- $O(\lg n)$ for plane graphs [Eppstein et al. - JCSS 1992]

$\boxed{\text{OPEN: } O(\lg n) \text{ for general graphs?}}$

- $O(\lg n (\lg \lg n)^3)$ update, $O(\lg n / \lg \lg \lg n)$ query [Thorup - STOC 2000]
- $O(\lg^2 n)$ update, $O(\lg n / \lg \lg n)$ query [Holm, de Lichtenberg, Thorup - JACM 2001]
  → TODAY
- $O(x \lg n)$ update $\Rightarrow \Omega(\lg n / \lg x)$ query [Demaine & Pătraşcu - STOC 2004/SICOMP 2006]
  $O(x \lg n)$ query $\Rightarrow \Omega(\lg n / \lg x)$ update
  → LECTURE 6

→ both "match" lower bound trade-off

$\boxed{\text{OPEN: } o(\lg n) \text{ update \& polylg } n \text{ query?}}$

- $O(\sqrt{n})$ worst-case update, $O(1)$ query
  [Eppstein, Galil, Italiano, Nissenweig - JACM 1997]

$\boxed{\text{OPEN: polylg worst-case update \& query}}$

Dynamic connectivity in $O(\lg^2 n)$ [Holm et al. - JACM 2001]

- store spanning forest with Euler tour trees
- hierarchically divide connected components

$\Rightarrow O(\lg n)$ levels of spanning forests, each Euler tour trees

- level (charging mechanism) of edge starts at $\lg n$, only decreases $\to \emptyset$
- G_i = subgraph of edges at level $\le i \Rightarrow G_{\lg n} = G$

  INVARIANT 1: every conn. component of G_i has $\le 2^i$ vxs.

- F_i = spanning forest of $G_i \Rightarrow F_{\lg n}$ = desired SF of G

  INVARIANT 2: $F_0 \subseteq F_1 \subseteq \cdots \subseteq F_{\lg n}$ i.e. $F_i = F_{\lg n} \cap G_i$

  i.e. $F_{\lg n}$ is min. spanning forest w.r.t. level

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

- add e to v & w adjacency lists
- level(e) $\leftarrow \lg n$
- if v & w disconnected in $F_{\lg n}$: add e to $F_{\lg n}$

$\Rightarrow O(\lg n)$

Queries in $O(\lg n / \lg \lg n)$:

- modify branching factor of $T_{\lg n}$ to $\Theta(\lg n)$ [B-tree]

$\Rightarrow O(\lg^2 n / \lg \lg n)$ to update (depth · branching)

& $O(\lg n / \lg \lg n)$ to findroot (depth)

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

- remove e from v & w's adjacency lists
- if e is in $F_{\lg n}$:
  - delete e from $F_{level(e)}, \dots, F_{\lg n}$
  - look for replacement edge to reconnect v & w
    - can't be at level < level(e) by MSF Invariant 2
    - find min. possible level ⇒ preserve Invariant 2

  for $i = level(e), \dots, \lg n$:
    - let T_v, T_w be trees of F_i with v, w resp.
    - relabel so that $|T_v| \le |T_w|$ (vertex count augment)
    - Invariant 1 ⇒ $|T_v| + |T_w| \le 2^i \Rightarrow |T_v| \le 2^{i-1}$

    ⇒ can afford to push all of T_v down to level $i-1$
    - for each edge (x, y) at level i with x in T_v:
      - if y is in T_w:

        add (x, y) to $F_i, F_{i+1}, \dots, F_{\lg n}$

        stop
      - else: $level(x, y) \leftarrow i-1$ charge

⇒ $O(\lg^2 n + \#\text{charges} \cdot \underline{\lg n})$

- each inserted edge charged $\le O(\lg n)$ times
- Euler tour tree augmentation:
  - subtree sizes to test $|T_v|$ vs. $|T_w|$ in $O(1)$
  - for each node v in tree of F_i: does v's subtree contain any nodes incident to level-i edges?

    ⇒ can find next level-i edge incident to $x \in T_v$ in $\underline{O(\lg n)}$ time (successor, jumping over empty subtrees)

Simpler dynamic connectivity problems:

- incremental: insertions only
  - $O(\alpha)$ amortized via union-find
  - worst case: $\Theta(x)$ updates $\Rightarrow \Theta(\lg n / \lg x)$ queries
- decremental: deletions only
  - $O(m \lg n + n \operatorname{polylg} n)$ for m updates, $O(1)$ query [Thorup – JACM 1999]

Other dynamic graph problems:

- minimum spanning forest (MST/conn. comp., as dynamic tree)
  - $O(\lg^4 n)$ update [Holm, de Lichtenberg, Thorup – JACM 2001]
  - worst case: $O(\sqrt{n})$ update [Eppstein et al. – JACM 1997]
  - plane graphs: $O(\lg n)$ [Eppstein et al. – JACM 1992]
- bipartiteness: is graph 2-colorable?
  - reducible to MSF
- planarity testing: insert e or report planarity violation
  - $O(n^{2/3})$ [Galil, Italiano, Sarnak – JACM 1997]
  - fixed embedding (plane): $O(\lg^2 n)$ [Eppstein et al. – JACM 1997]
  - incremental: $O(m\,\alpha(m,n)+n)$ [la Poutre – STOC 1994]

$\boxed{\text{OPEN: testing for any fixed minor?}}$

k-connectivity: vertex or edge

- disjoint paths between pairs of vertices:
  - $O(\text{polylg } n)$ for $k=2$ [Holm et al. — JACM 2001]
  - planar decremental: $O(\lg^2 n)$ for 3-edge-conn. [Giammaresi & Italiano — Algorithmica 1996]
  - worst case: [Eppstein et al. — JACM 1997]
    - $O(\sqrt{n})$ for 2-edge-conn.
    - $O(n)$ for 2-vertex-conn. & 3-vertex-conn.
    - $O(n^{2/3})$ for 3-edge-conn.
    - $O(n\,\alpha(n))$ for $k=4$
    - $O(n \lg n)$ for $O(1)$-edge-conn.

OPEN: polylg n for $k=O(1)$? $k=\text{polylg } n$?

- whole graph ($\sim$ min cut = max flow)
  - $O(\sqrt{n}\,\text{polylg } n)$ for $O(\text{polylg } n)$-edge-conn. (& min cut up to that size) [Thorup — STOC 2001]

OPEN: polylg n for $k=O(1)$? $k=\text{polylg } n$?

Dynamic directed graphs:

Transitive closure: is there a path from v to w?

- bulk update: insert/delete vertex & incident edges
- $O(n^2)$ amortized bulk update, $O(1)$ worst-case query [Demetrescu & Italiano - FOCS 2000; Roditty - SODA 2003]
- same, worst case [Sankowski - FOCS 2004]
- optimal if storing transitive closure matrix explicitly

OPEN: $o(n^2)$ update worst case?

- $O(m\sqrt{n} \cdot t)$ am. bulk update, $O(\sqrt{n}/t)$ w.c. query, $t = O(\sqrt{n})$ [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 with update · query = $O(m \cdot n)$ or $O(n^2)$

- 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: shortest-path weight $v \to w$?

- $\tilde{O}(n^2(\lg n + \lg^2(1 + m/n)))$ 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 for undirected graphs?

- $\tilde{O}(n^{2.75})$ w.c. update, $O(1)$ query [Thorup - STOC 2005]
- unweighted: $O(m\sqrt{n} \cdot \text{polylg } n)$ am. update, $O(n^{3/4})$ w.c. query [Roditty & Zwick - ESA 2004]
- undirected, unweighted, & $(1+\varepsilon)$-approx.: $O(\sqrt{m}\,n \cdot t)$ am. update, $O(\sqrt{m}/t)$ w.c. query, $t = O(\sqrt{n})$ [Roddity & Zwick - FOCS 2004]