November 3, 2023 10:51 AM

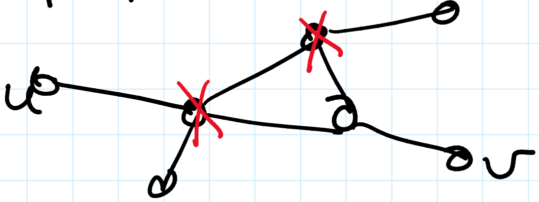
Problem (Dynamic Connectivity with Vertex Updates)

Dynamic Subgraph Conn.

maintain undir graph $G=(V,E)$ and subset $S \subseteq V$, "active" vertices

- (re)insert v to S (turn "on")
- delete v from S ("off")
- query: are u & v connected in subgraph induced by S?

$(n=|V|, m=|E|)$



Rmk - $O(\deg(v) \cdot \log^2 n)$ amort. time by doing $\deg(v)$ # edge updates

but deg can be large!

C'02: $O(m^{0.94})$ amort. update time

$O(m^{1/3})$ query time

matrix mult.

C., Patrascu, Roditty '08: $\tilde{O}(m^{2/3})$ amort. update,

$\tilde{O}(m^{1/3})$ query time

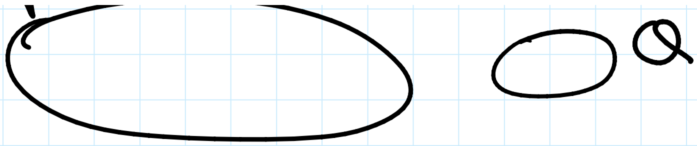
---

idea 1 - periodic rebuilding, after every q updates

decompose, $S = P \cup Q$

P: large, deletion-only

Q: small, $|Q| \le q$. insert/delete allowed.

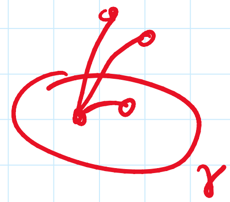


after q updates, reset $P = S$, $Q = \emptyset$.

idea 2. low deg $\Rightarrow$ good
high deg: not too many

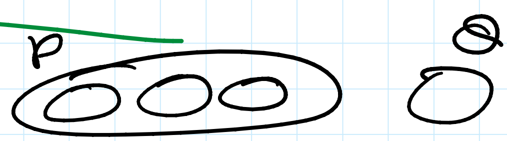
Call connected component γ of P high
if $\deg(\gamma) > \Delta$.
low else

$= \sum_{v \in \gamma} \deg(v)$



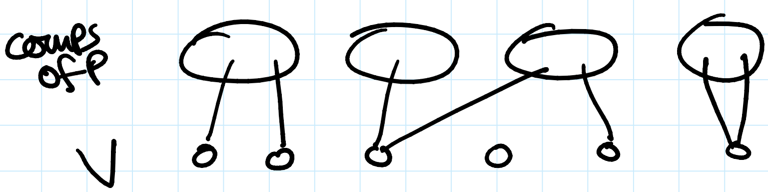
$\Rightarrow$ # high comps $\leq O\left(\frac{m}{\Delta}\right)$

Data Structure!



1. comps of P in dyn conn DS with edge updates
2. bipartite multigraph Γ
   between V & comps of P

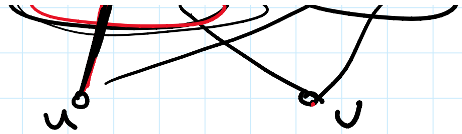
preproc $\tilde{O}(m)$



3. $\forall u, v \in V$,
   $C_{low}[u,v]$ = # low comps of P adj to u & v in Γ

preproc time $\tilde{O}(m\Delta)$

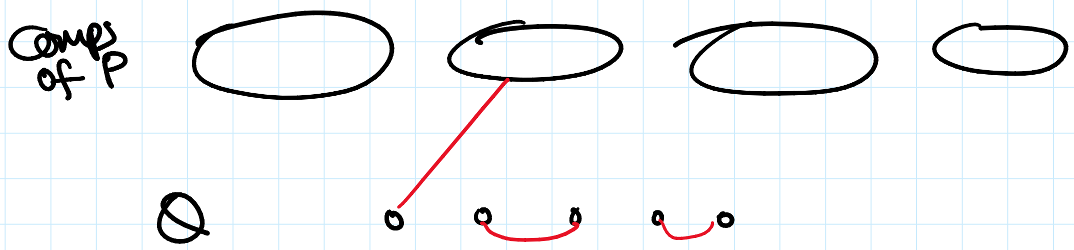
$\tilde{O}(m\Delta)$



4. define "proxy graph" G^*:

vertices are (comps of P) $\cup$ Q

maintained in dyn conn DS with edge updates

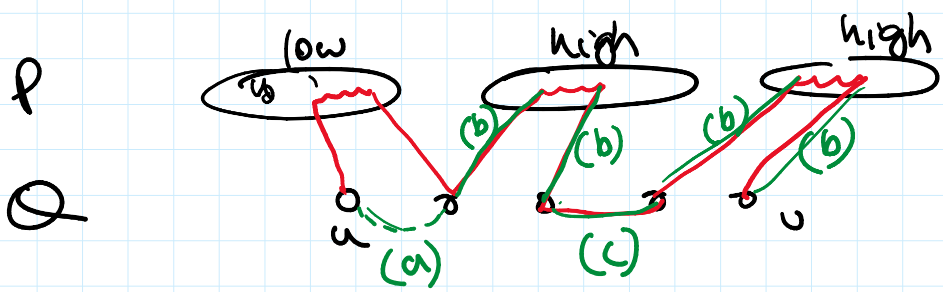


a) whenever $u, v \in Q$ & $C_{low}[u,v] > 0$, add uv to G^*

b) whenever $u \in Q$ is adj to high comp γ in P, add $u\gamma$ to G^*.

c) whenever $u, v \in Q$ & $uv \in E$, add uv to G^*.

Obs $\forall u, v \in Q$, u & v connected in subgraph of G induced by S $\iff$ u & v conn. in G^*.



Query(u,v): if $u, v \in Q$, done ✓

if u, v in same comp of P done ✓

if u in high comp, pick any adj vertex $u' \in Q$ in G^*

if u in low comp,

pick any adj vertex $u' \in Q$ in G

if u in low comp,
find adj vertex $u' \in Q$ by brute force ← $O(\Delta)$ time

replace u with u'
same for v

$\Rightarrow \underline{\tilde{O}(\Delta)}$

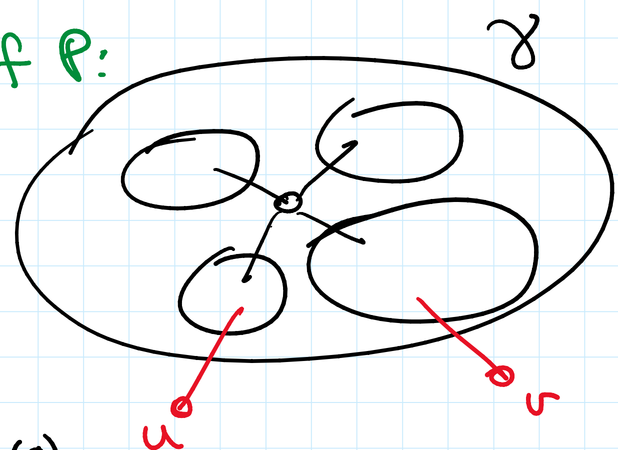
Insert/delete v to Q:

$O(q)$ edge updates of type (a), (c)

$O(\underline{m/\Delta})$ edge updates of type (b)

$\Rightarrow \tilde{O}(q + \frac{m}{\Delta})$.

Delete v from low comp γ of P:

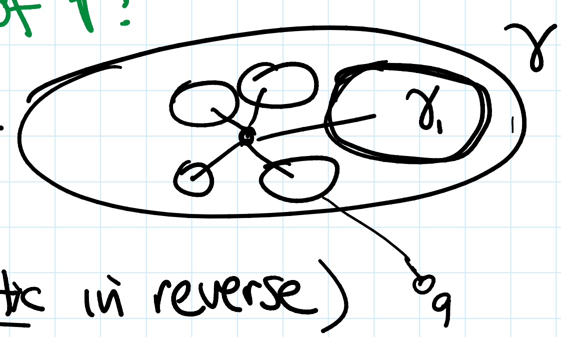


Split γ into subcomps.

$C_{low}[\cdot,\cdot]$: update $O(\Delta^2)$ entries

$\Rightarrow O(\Delta^2)$ edge updates of type (a).

Delete v from high comp γ of P:



Split γ into subcomps $\gamma_1, \ldots, \gamma_\ell$
Say γ_1 largest

(use weighted union heuristic in reverse)

Split $\gamma_2, \ldots, \gamma_\ell$ from γ_1

Split $v_2, \dots, v_\ell$ from v_1

$\Rightarrow$ # edge updates of type (b)

$$O(\deg(\gamma_2) + \dots + \deg(\gamma_\ell))$$

$\xRightarrow{\text{total}}$ $O(m \log m)$.

update $G_{low}[\cdot, \cdot]$:

$$O((\deg(\gamma_2) + \dots + \deg(\gamma_\ell)) \cdot \Delta)$$

$\xRightarrow{\text{total}}$ $O(m \Delta \log m)$.

Rebuilding cost: $\tilde{O}\left(\frac{m\Delta}{q}\right)$ amort.

$\Rightarrow$ Overall amort update time

$$\tilde{O}\left(\frac{m\Delta}{q} + q + \frac{m}{\Delta} + \Delta^2\right)$$

Set $q = \Delta^2$: $\tilde{O}\left(\frac{m}{\Delta} + \Delta^2\right)$

Set $\Delta = m^{1/3}$: $\boxed{\tilde{O}(m^{2/3})}$ amort update ✓

$\boxed{\tilde{O}(m^{1/3})}$ query