* Quiz + Discussion (2-3 PM)

---

* Student talk - Decremental SSSP

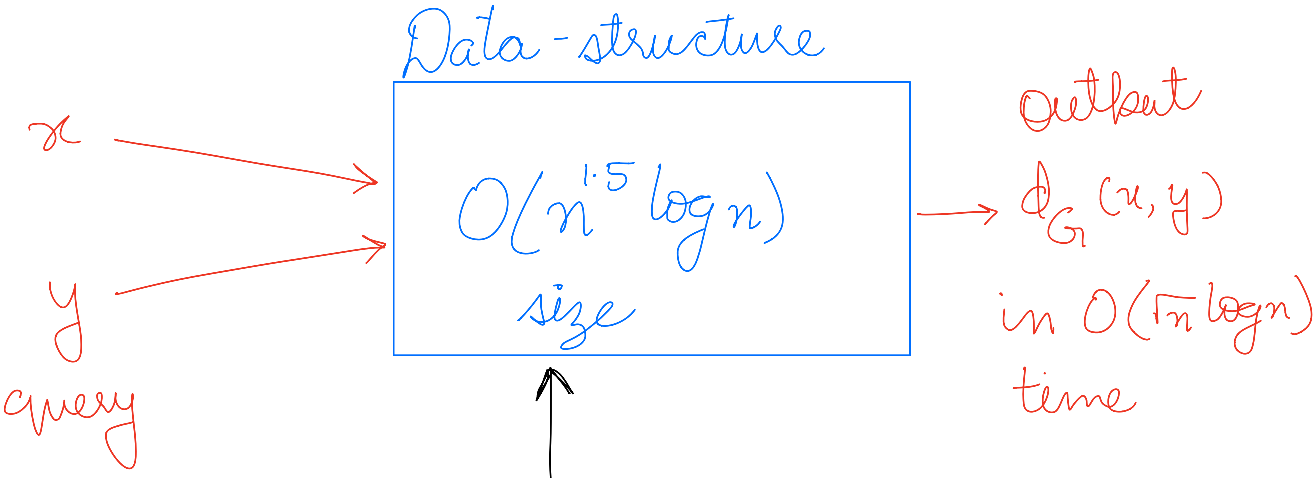
Application of decremental SSSP:

Problem statement:

Maintain a oracle (data-structure) of $O(n^{1.5} \log n)$ size decrementally (i.e. for a graph undergoing edge deletions), so that:

(i) after every edge deletion, the oracle can be updated quickly (in amortized $O(n^{1.5} \log n)$ time.)

(ii) given any query pair (x, y) the oracle reports $d_G(x, y)$ in $O(\sqrt{n} \log n)$ time.



On deletion of edge e
data structure can be update in $O(n^{1.5} \log n)$ time

We mimic the algorithm taught in last class:

Data-Structure

(1) For each $v \in V$
we maintain out-$BFS_{\sqrt{n}}(v)$

(decrementally)

(2) $R \leftarrow$ Random set of size "$16\sqrt{n} \log n$"

For each $v \in R$,
we maintain in-bfs(v) & out-bfs(v)

Total update time:

$$|V| \cdot O(m\sqrt{n}) + |R| \cdot O(mn)$$

$$= O(mn\sqrt{n} \log n)$$

$\Rightarrow$ Amortized update time
$= O(n\sqrt{n} \log n)$

How to find $d_G(x,y)$ for query pair $(x,y) \in V \times V$?

(a) If $y \in \text{OUT-BFS}_{\sqrt{n}}(x)$	(b) If $y \notin \text{OUT-BFS}_{\sqrt{n}}(x)$
Report $d_G(x,y)$ from tree of x in $O(1)$ time.	$\Rightarrow d_G(x,y) > \sqrt{n}$ x-y shortest path  Report $\min_{r \in R} \{ d_G(x,r) + d_G(r,y) \}$

worst case query time

$$= O(|R|) = O(\sqrt{n} \log n)$$

Question: How to incrementally / decrementally maintain a +2 additive spanner?

Recall. (lecture 2)

Algorithm to compute "+2" additive spanner.

1. Initialize $H = (V, E_H = \emptyset)$
2. $R \leftarrow$ uniform random subset of V of size $\Theta(\sqrt{n})$.
3. For each $w \in R$:
   Add edges of $BFS(w, G)$ to E_H.
4. For each $v \in V(G)$ with $deg(v, G) \leq 4\sqrt{n}\log n$:
   Add to E_H all edges incident to v in G.
5. Return H

In order to dynamically maintain +2 additive spanner, we just need to

1. store in H edges incident to all vertices of current degree $\leq \sqrt{n}$.
2. dynamically maintain $BFS(r)$, for $r \in R$

Total time will be $= O(m \cdot n \cdot |R|)$

$= O(m \cdot n \cdot \sqrt{n} \log n)$

in incremental / decremental setting.