Known Results:

	PROBLEM	TOTAL TIME
Last 2 classes →	Incremental/Decremental maintenance of BFS(s) for a given vertex $s \in V(G)$	$O(mn)$
H.W. →	Incremental/Decremental maintenance of first "k"-levels of BFS('s)	$O(mk)$

Recall: In static setting, time to solve APSP (all pairs shortest path) is $O(mn)$.

≤ 0.1

Problem: We now consider $(1+\varepsilon)$-approx APSP in incremental/decremental setting

Goal:

x, y → **Data Structure D** → $(1+\varepsilon)$-approximation of $d_G(x,y)$ in $O(\log_2 n)$ time

Total time to update this should be $O\left(\frac{mn}{\varepsilon}\log^2(n)\right)$

We will present result for incremental setting

G_i = graph at time $t=i$ (with i edges)

G_0 = initial graph (0 edges)

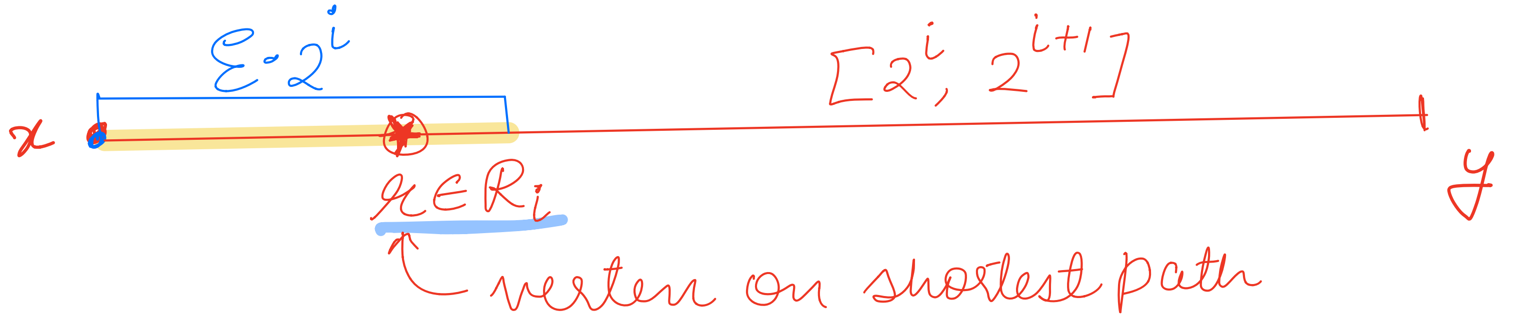
G_m = final graph (m edges)

$\boxed{d_{G_t}(x,y)}$ = x to y distance at time t (i.e. in graph G_t)

can lie in range:

$[1,2]$ $[2,4]$ $[4,8]$ .... $[2^i, 2^{i+1}]$ ....

(*) Suppose: $2^i \le d_{G_t}(x,y) \le 2^{i+1}$



$R_i \equiv$ random set of size $\Theta\left(\frac{n \log n}{\varepsilon \cdot 2^i}\right)$

Let $c \in R_i$ be:

"c" $= c(x, R_i) =$ vertex in R_i closest from x

Note: $d_{G_t}(x,c) \le d_{G_t}(x,r) \le \varepsilon \cdot 2^i$

- $d_{G_t}(c,x) \le \varepsilon \cdot 2^i \le 2^i$

- $d_{G_t}(c,y) \le d_{G_t}(c,x) + \underbrace{d_{G_t}(x,y)}_{\in [2^i, 2^{i+1}]} \le 3 \cdot 2^i$.

How to find $\hat{d}_{G_t}(x,y)$ $\left(= 1+2\varepsilon \text{ approximation to } d_G(x,y)\right)$:

Step 1: Find c = vertex in R_i closest to x.

Step 2: Output $\hat{d}_{G_t}(x,y) := d_{G_t}(x,c) + d_{G_t}(c,y)$.

Lemma: $\hat{d}_{G_t}(x,y) \leq (1+2\varepsilon)\, d_{G_t}(x,y)$.

Proof:

$$\hat{d}_{G_t}(x,y) := d_{G_t}(x,c) + d_{G_t}(c,y)$$

$$\leq d_{G_t}(x,c) + d_{G_t}(c,x) + d_{G_t}(x,y)$$

$$\leq 2\cdot\varepsilon\cdot 2^i + d_{G_t}(x,y)$$

$$\leq (1+2\varepsilon)\, d_{G_t}(x,y) \quad \square$$

Note: We use the fact that

(i) G is undirected (ii) $d_{G_t}(x,c) \leq \varepsilon\cdot 2^i$

We incrementally maintain:

* $\forall c \in R_i$,

$$BFS(c, \text{depth} \leq 3 \cdot 2^i)$$

Analysis of time to maintain this:

$$|R_i| = \Theta\left(\frac{n \log n}{\varepsilon 2^i}\right)$$

$\forall c \in R_i$, we maintain $BFS(c, \text{depth} \leq 3 \cdot 2^i)$

→ Total time for incremental maintenance $= O(|R_i| \cdot m \cdot 3 \cdot 2^i)$

$$= O\left(\frac{n \log n}{\varepsilon \cancel{2^i}} \cdot m \cdot 3 \cdot \cancel{2^i}\right)$$

$$= O\left(mn \cdot \frac{\log n}{\varepsilon}\right)$$

* For all indices time will be $= O\left(mn \cdot \frac{\log^2 n}{\varepsilon}\right)$

Algorithm to find $\hat{d}_{G_t}(x,y)$:

(*) For $i = 1$ to $\log_2 n$

- Find $c = c(x, R_i)$ ← How to find c? We will see later.
- check $x, y \in BFS(c, \text{depth} \leq 3\cdot 2^i)$
- $temp_i = d_{G_t}(x,c) + d_{G_t}(c,y)$

(*) Output $\min\{temp_1, temp_2, \ldots temp_{\log_2 n}\}$

Correctness Proof:

If $2^i \leq d_{G_t}(x,y) \leq 2^{i+1}$

$$\Rightarrow d_{G_t}(x,c) + d_{G_t}(c,y) \leq (1+2\varepsilon)\, d_{G_t}(x,y)$$

i.e. $temp_i \leq (1+2\varepsilon)\, d_{G_t}(x,y)$

$$\Rightarrow \min\{temp_1, \ldots, temp_{\log n}\} \leq (1+2\varepsilon)\, d_{G_t}(x,y)$$

QUERY TIME:

$O(\log n)$ to find minimum, if we can find "c" in $O(1)$ time.

All that remains is to efficiently maintain:

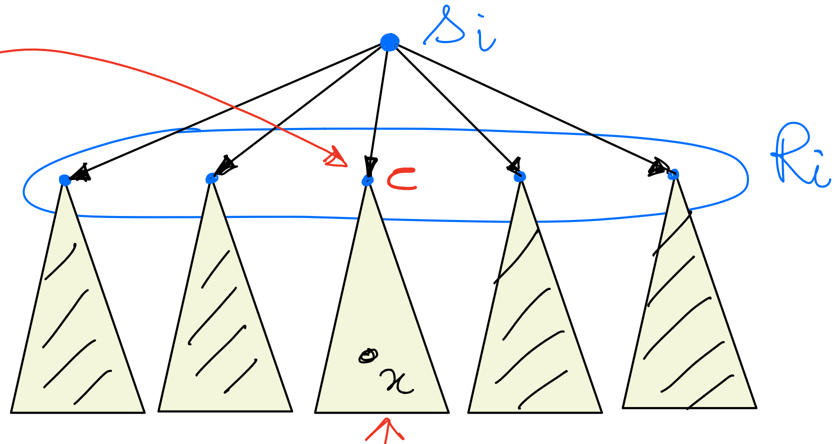
"c" $= \underbrace{c(x, R_i)}$ = vertex in R_i closest from x

We achieve this using a simple trick:

For each $i \leq \log_2 n$, we do the following:

Maintain BFS(supernode $= R_i$)

To do this, add dummy (new) node s_i to G,
and connect s_i to all vertices in R_i



This is "$c(x, R_i)$"

if

x lies in this subtree

(*) Whenever x changes its parent during an edge insertion it can learn the "new c" from its "new-parent" in BFS(supernode $= R_i$).

So, time to find $c = c(x, R_i)$ while answering query $\hat{d}_{G_t}(x,y)$ will be just $O(1)$.

Description of data-structure for maintaining "c"

1. R_i = Random set of $O\left(\frac{n \cdot \log n}{\varepsilon 2^i}\right)$ vertices.
2. $G_{(i)}$ = copy of G with dummy source s_i.
3. Connect s_i to all vertices in R_i.
4. Maintain $BFS_{G_{(i)}}(s_i)$.

Total time to maintain this $= O(m \cdot n \cdot \log n)$ } bcoz $1 \leq i \leq \log_2 n$

Summary of Result so far:

* Total time to maintain data structure

$$= O\left(mn \cdot \frac{\log^2 n}{\varepsilon}\right) + O(mn \log n)$$

$$= O\left(mn \frac{\log^2 n}{\varepsilon}\right)$$

* Query time $= O(\log n)$

Question:

How to answer queries in $O(\log \log n)$ time?

Can we use idea of binary search?

Proposition: Suppose $i \in [1, \log n]$ satisfy $d_{G_t}(x, y) \in [2^i, 2^{i+1}]$. Then for any index j we can verify in $O(1)$ time following:

* Whether or not $j \leq i$.
* Whether or not $j \geq i$.

Proof: For $j = 1$ to $\log n$

Let c_j = vertex in R_j closest to x

Recall $|R_j| = \Theta\left(\frac{n \log n}{\varepsilon 2^j}\right)$

$$\Rightarrow d_{G_t}(x, c_j) \leq \varepsilon \cdot 2^j$$

Observe: $d_{G_t}(c_j, y) \in \left[2^i - \frac{2^j}{10},\ 2^{i+1} + \frac{2^j}{10}\right]$

Proof:

- $d_{G_t}(c_j, y) \leq d_{G_t}(c_j, x) + d_{G_t}(x, y)$
  $\leq 2^{i+1} + \varepsilon 2^j$

- $d_{G_t}(c_j, y) \geq d_{G_t}(x, y) - d_{G_t}(c_j, x)$
  $\geq 2^i - \varepsilon 2^j$

- $\varepsilon \leq 0.1$

Now,

(i) If $j \nleq i$, then:

$$d_{G_t}(c_j, y) \geq 2^i - \varepsilon 2^j$$
$$\geq 2^{j+1} - (0.1) 2^j \ngtr (1.5) 2^j$$

(ii) If $j \ngtr i$, then:

$$d_{G_t}(c_j, y) \leq 2^{i+1} + \varepsilon 2^j$$
$$\leq 2^j + (0.1) 2^j \nleq (1.5) 2^j$$

Now, we can check if $d_{G_t}(c_j, y)$ is less than or greater than $(1.5)(2^j)$ because we maintain $BFS(c_j, \text{depth} \leq 3 \cdot 2^j)$. Thus,

- $j \leq i$ only when $d_{G_t}(c_j, y) \geq (1.5) 2^j$
- $j \geq i$ only when $d_{G_t}(c_j, y) \leq (1.5) 2^j$

So, the claim follows.