Lecture 19:

LCAs for Spamers
Graph Spanners:

Given $G = (V, E)$

def. $k$-spanner is subgraph $H = (V, E')$ s.t.

$\forall u, v \in V$ \quad $\text{dist}_H(u, v) \leq k \cdot \text{dist}_G(u, v)$

Known $\forall G$, $\exists (2k-1)$-spanner with $O(n^{1+k})$ edges $\exists 3$-spanner with $O(n^{3/2})$ edges

Optimal? yes for $k=2, 3, 5$

Erdos girth conjecture $\Rightarrow$ yes for all $k$

Equivalent Characterization:

$\forall (u, v) \in E$, $\exists$ path from $u$ to $v$ in $H$ of length $\leq K$

Question: LCA which given graph $G$ provides queries to spanner $H$?

How is $G$ given? Assume following probes:

neighbor: given $u, i$ output $i^{th}$ nbr of $u$

adjacency: given $(u, v)$ output whether $(u, v) \in G$

output: $j$ if $(u, v) \in G$ + "no" if $(u, v) \notin G$

degree: given $u$ output $\deg(u)$
LCA for 3-spanner with $\tilde{O}(n^{3/2})$ edges + $O(n^{3/4})$ time/query.

First, a thought:

Pick centers randomly
if $u, v$ both connected
to same center, can
delete edge $(u, v)$

$\text{dist}(u, v) = 1$ but $\text{dist}_H(u, v) = 2$ (ok, since $k = 3$)

but: will we delete enough edges this way?
Can we figure out that $u, v$ connected to same center
in sublinear time?

Today: will assume max degree is $n^{3/4}$
- still nontrivial
- general case builds on ideas today
Global construction of 3-spanners with $\tilde{O}(n^{3/2})$ edges

[Baswana Sen 07]

Note: ave degree $= n^{1/2}$

Construction of $H$: (not sublinear time)

- Pick $S \subseteq V$ s.t. $|S| = \Theta(\sqrt{n} \cdot \log n)$ \leftarrow each node tosses coin with prob $\Theta(\frac{\log n}{\sqrt{n}})$

"cluster Centers" \leftarrow each one defines a "cluster"

- w.h.p., $\forall u \in V$ s.t. $u$ has degree $\geq \sqrt{n}$, then $u$ adjacent to at least one $v \in S$

$u$ chooses one $v \in S$ (arbitrarily) to be its "cluster center"

$\tilde{O}(n^{3/2})$ total

- Constructing $H$:
  1. if $u$ low degree ($\leq \sqrt{n}$), add all edges $(u,v)$
  2. if $u$ high degree ($\geq \sqrt{n}$), add edge to its cluster center
  3. if $u$ high degree ($\geq \sqrt{n}$), add one edge to every adjacent cluster

$\leq n \cdot \sqrt{n}$

$\leq n \cdot \log n$

$\leq n \cdot \log n$ at clusters
Example:

Clusters

Stretch?

- For $u, v$ in same cluster, both $u$ and $v$ keep edge to center $c$ rolled $H(u, v) = 2$

- For $u, v$ in different clusters:

\[
\text{if } (u, v) \notin H \text{ then must have kept some other edge } (u, w) \text{ s.t. } w \in v's \text{ cluster.}
\]

So either $w = c_v$ or $(w, c_v) \in H$

\[
\Rightarrow (u, w), (w, c_v), (c_v, v) \in H
\]

\[
\Rightarrow \text{dist}_H(u, v) = 3
\]
Local Algorithm for constructing H:

Given $(u,v) \in E$, is $(u,v) \in H$?

Rule (1): if $u$ or $v$ low degree, yes! 2 degree probes $\checkmark$

Rule (2): if $v$ is $u$'s center (or if $u$ is $v$'s center)

Rule (3): if $(u,v)$ is "chosen" edge from $u$ to $v$'s cluster (or $v$ to $u$'s cluster)

How do we know?

Naive idea: "First center Attempt"

$u$ chooses $1^{st}$ center on its incidence list

$u \rightarrow w_1, w_2, ..., w_k, \ldots, w_{m-1}, w_m, \ldots$

$u$ chooses $1^{st}$ connection to each cluster in incidence list

$u \rightarrow w_1, w_p, ..., w_k, \ldots, w_l, \ldots, w_m$
Implementing rule 2:

- On query \((u,v)\): if \(v\) the chosen center of \(u\)?
  - Check if \(v\) is a center (check \(v\)'s coin toss)
  - Check if any node preceding \(v\) on \(u\)'s incidence list is a center

Runtime: \(O(\text{max degree})\)

Better runtime: \(O(n)\) by observation

Implementing rule 3:

- On query \((u,v)\): does \(v\) introduce \(u\) to a new cluster?
  - Find \(v\)'s cluster center \(C_v\) \(O(n)\)
  - Check all nbns of \(C_v\) + check if come earlier in \(u\)'s incidence list?

Not sublinear for \(\Delta = \Delta(n)\) (regime of interest)
**Improved Plan: "Multiple Centers"**

**Rule 2:** \( u \) chooses all centers in first \( \sqrt{n} \) locs of incidence list \( C_u = \{ v \mid v \text{ is in first } \sqrt{n} \text{ locs of } u \text{'s incidence list} \} \)

**Observation:** \( \forall u \text{ s.t. } \deg(u) \geq \sqrt{n}, 1 \leq |C_u| \leq \log n \)

How does this change things?

- degree from Rule 2 "keep all edges between \( u \) and \( C_u \)"
  - \( O(\log n) \) per node
  - \( \Rightarrow O(n \log n) \) total [before it was \( O(n) \) total]

- Verifying if \( v \in C_u \):
  - adjacency probe \((u, v)\) returns \( v \)'s loc in \( u \)'s list
  - in one step
  - check if \( v \) is a center by looking at random \( \sqrt{n} \)s

**SAVINGS!!!**

- computing \( C_u \):
  - check 1st \( \sqrt{n} \) locs in \( u \)'s list
to see which are centers

![](image)
Rule 3: \( u \) chooses first edge \( v \) which introduces \( u \) to \( v \)'s cluster

How to determine?

1. Compute \( C_v \): \( \tilde{\nu}n \) neighbor probes

2. For each \( w \in C_v \), test if \( v \) "introduces" \( w \) to \( u \):
   
   For each nbr \( x \) of \( u \) until reach \( v \):
   
   Find \( C_x \)
   
   Cross off \( C_x \cap C_v \)
   
   If any \( w \in C_v \) not crossed off
   
   Then keep \( (w,v) \) in \( H \)
   
   Else discard \( (w,v) \)

Total: \( O(\Delta \cdot \tilde{\nu}n \cdot \log n) \)

Bad!!
Smarter method to determine if \( v \) "introduces" closer to \( u \): 

- Compute \( C_v \)

- For each \( n \)br \( x \) of \( u \) up to \( v \):
  - \( \text{deg}(u) \) probes
    - For each \( w \in C_v \),
      - if \( w \) is center of \( x \)
        - cross \( w \) off
    - If any \( w \in C_v \) not crossed off
      - Keep \( (u,v) \) in \( H \)
    - else discard.

Total:
\[
\sqrt{n} + \text{deg}(u) \times \log n \times 1
\]
\[
= O(\text{deg}(u) \cdot \log n)
\]

If \( \max_{u} \text{deg}(u) \leq n^{3/4} \), we are done!!