Lecture 19:

LCAs for spanners

Graph Spanners:

Given $G=(V,E)$

• same nodes
• $E' \subseteq E$

<u>def.</u> **k-spanner** is subgraph $H=(V,E')$ s.t.

$\forall u,v \;\; dist_H(u,v) \leq k \cdot dist_G(u,v)$

<u>Known</u> $\forall$ G, $\exists$ $(2k-1)$-spanner with $O(n^{1+\frac{1}{k}})$ edges $\Big\}$ e.g. $k=2$ $\exists$ 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:

$\forall (u,v) \in H$, $\exists$ path from u to v in H of length $\leq k$ $\Big\}$ So, whenever we omit an edge (u,v) we will make sure a path of length $\leq k$ remains between u + v

<u>Question</u>: LCA which given graph G provides queries to spanner H?

<u>How is G given?</u> Assume following probes:

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

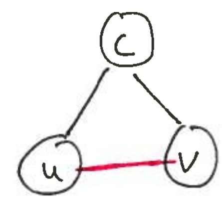
adjacency: given (u,v) output whether $(u,v) \in G$
<u>or</u>
output j if $(u,v) \in G$ + "no" if $(u,v) \notin G$

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)



$\text{dist}_G(u,v) = 1$ 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 $\approx n^{1/2}$

Construction of H: (not sublinear time)

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

  "cluster centers" ⇐ each one defines a "cluster"

- whp, $\forall\ u \in V$ s.t. u has degree $\geq \sqrt{n}$, then u adjacent to at least one center $v \in S$ } useful observation *

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

- Constructing H:
  1. if u low degree ($< \sqrt{n}$), 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 <u>one</u> edge to every adjacent cluster

#edges

$< n \cdot \sqrt{n}$

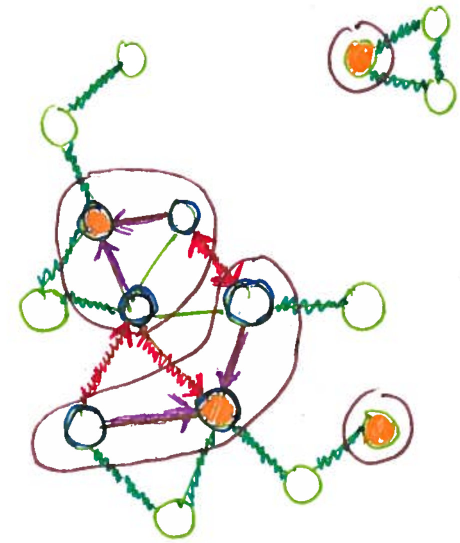
$< n \cdot 1$

$< n \cdot \underbrace{\sqrt{n} \cdot \log n}_{\#\text{ clusters}}$

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

Example:

Clusters



low degree

high degree

cluster center

edges added due to

rule 1: low degree

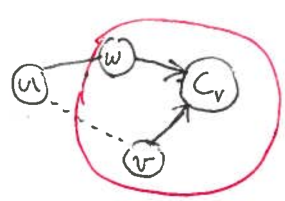
rule 2: cluster center

rule 3: adjacent cluster

(directed edges only indicate who made the choice, actual edges are all Undirected)

Stretch?

- for u, v in same cluster, both $u + v$ keep edge to center c $\Rightarrow \text{dist}_H(u,v) = 2$
- for u, v in different clusters:



if $(u,v) \notin H$ then must have kept some other edge (u,w) s.t. w in v's 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 $(u,v) \in G$, is $(u,v) \in H$?

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

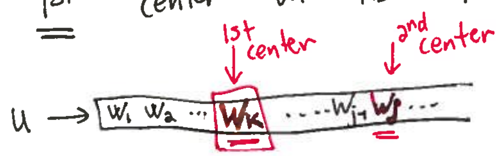
Rule (2): if v is u's center (or if u is v's center)

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

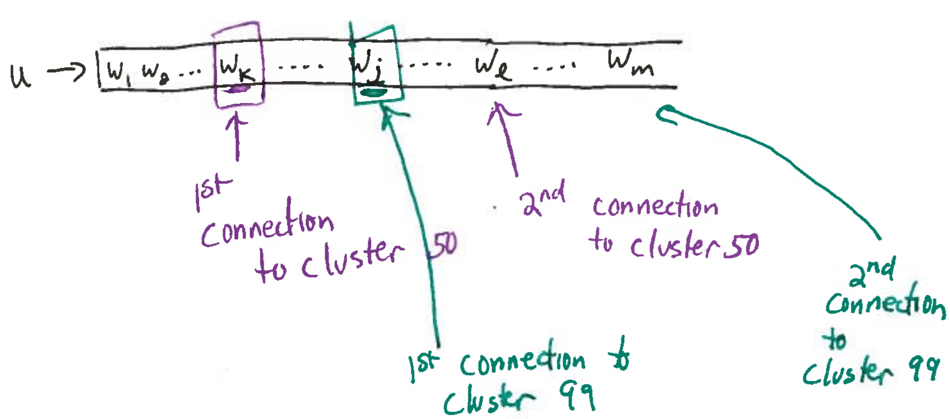
How do we know?

Naive idea: "First center Attempt"

u chooses 1st center on its incidence list



u chooses 1st connection to each cluster in incidence list



Implementing rule 2:

on query (u,v): if v the chosen center of u?

need to also check if u is chosen center of v

- 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: $O(\sqrt{n})$ by observation $*'$

Implementing rule 3:

on query (u,v): does v introduce u to a new cluster?

- find v's cluster center C_v $O(\sqrt{n})$
- check all nbrs of C_v + check if come earlier in u's incidence list?

need to make sure C_v is w's center

$O(\Delta \cdot (1+\sqrt{n}))$

$\forall w \in N(C_v)$ adjacency probe (u,w) returns locn in u's list

not sublinear for $\Delta = \Omega(\sqrt{n})$ (regime of interest)

Improved Plan: "Multiple Centers"

Rule 2: u chooses <u>all</u> centers in first $\sqrt{n}$ locns of incidence list

$C_u = \{ v \mid v$ is in 1^{st} $\sqrt{n}$ locns of u's incidence list & v is a center$\}$

<u>Observation</u> WHP, $\forall u$ s.t. $\deg(u) \geq \sqrt{n}$, $1 \leq |C_u| \leq \log n$

$\boxed{\text{check this!}}$

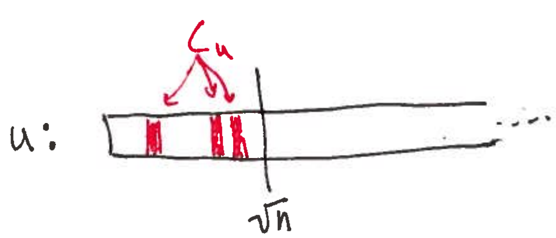
How does this change things?

- degree from rule 2 "keep all edges between u & C_u" is $O(\log n)$ 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 locn in u's list in one step
  - check if v is a center by looking at random ($)'s

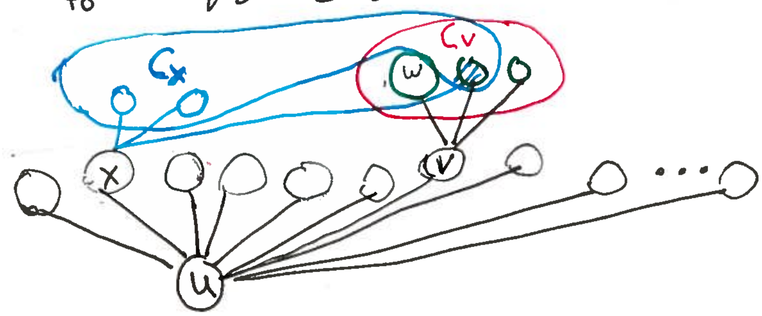
SAVINGS!! ⟹ 1 adjacency probe

- computing C_u:
  - check 1^{st} $\sqrt{n}$ locns in u's list to see which are centers



$\sqrt{n}$ neighbor probes

Rule 3: u chooses first edge v which introduces u to v's cluster



How to determine?

- compute C_v $\sqrt{n}$ neighbor probes
- for each $w \in C_v$, test if v "introduces" w to u:

1st attempt

For each nbr x of u until reach v: $\deg(u)$

Find C_x $\sqrt{n}$ nbr probes

cross off $C_x \cap C_v$ $O(\log n)$

If any $w \in C_v$ not crossed off

then keep (u,v) in H

else discard (u,v)

Total: $O(\underbrace{\Delta \cdot \sqrt{n} \cdot \log n}_{\text{bad!!}})$

Smarter method to determine if v "introduces" cluster to u:

- Compute C_v — $\sqrt{n}$ nbr probes
- For each nbr x of u up to v: — $\deg(u)$

  For each $w \in C_v$, — $O(\log n)$

  if w is center of x — 1 adjacency probe

  cross w off

If any $w \in C_v$ not crossed off

Keep (u,v) in H

else discard.

Total:

$\sqrt{n} + \deg(u) \times \log n \times 1$

$= O(\deg(u) \cdot \log n)$

If $\max_u \deg(u) \le n^{3/4}$, we are done!!