Fast Distance Queries

Laurent Viennot

MPRI - Theory of practical graph algorithms
Encoding a graph metric: distance oracles
Size $S$ vs query time $T$ tradeoff (sparse graphs, i.e. $m = O(n)$)
Encoding a graph metric: distance labelings

\[ S = n \cdot \text{DistLab}(n) \]

\[ d_G(u, v) \]
Distance labeling [Gavoille, Peleg, Pérennes, Raz '04]

Problem
Given a graph $G$ assign a label $L_u$ to each node $u$ s.t. for all $s, t$ $d_G(s, t)$ can be computed from $L_s$ and $L_t$.

Hub sets
Given a graph $G$, assign a hub set $H_u \subseteq V(G)$ to each node $u$, s.t. for all $u, v$ there exists $a \in H_u \cap H_v$ with $a \in P_{uv}$.

Distance labels: $L_u = \{(a, d(u, a)) : a \in H_u\}$
Distance query: $\text{Dist} (L_s, L_t) = \min_{a \in H_s \cap H_t} d(s, a) + d(a, t)$ in $O(|H_s| + |H_t|)$ time.

Covering property: for all $u, v \in V$, $H_u \cap H_v \cap I_{uv} \neq \emptyset$ where the interval $I_{uv}$ is the union of all $uv$-shortest paths.
Distance labeling [Gavoille, Peleg, Pérennes, Raz '04]

Problem
Given a graph $G$ assign a label $L_u$ to each node $u$ s.t. for all $s, t$ $d_G(s, t)$ can be computed from $L_s$ and $L_t$.

Hub sets
Given a graph $G$, assign a hub set $H_u \subseteq V(G)$ to each node $u$, s.t. for all $u, v$ there exists $a \in H_u \cap H_v$ with $a \in P_{uv}$.

Distance labels: $L_u = \{(a, d(u, a)) : a \in H_u\}$
Distance query: $\text{Dist} (L_s, L_t) = \min_{a \in H_s \cap H_t} d(s, a) + d(a, t)$ in $O(|H_s| + |H_t|)$ time.
Covering property: for all $u, v \in V$, $H_u \cap H_v \cap I_{uv} \neq \emptyset$ where the interval $I_{uv}$ is the union of all $uv$-shortest paths.
Distance labeling [Gavoille, Peleg, Pérennes, Raz '04]

Problem
Given a graph $G$ assign a label $L_u$ to each node $u$ s.t. for all $s, t \ d_G(s, t)$ can be computed from $L_s$ and $L_t$.

Hub sets
Given a graph $G$, assign a hub set $H_u \subseteq V(G)$ to each node $u$, s.t. for all $u, v$ there exists $a \in H_u \cap H_v$ with $a \in P_{uv}$.

Distance labels : $L_u = \{(a, d(u, a)) : a \in H_u\}$

Distance query : $\text{Dist} (L_s, L_t) = \min_{a \in H_s \cap H_t} d(s, a) + d(a, t)$ in $O(|H_s| + |H_t|)$ time.

Covering property : for all $u, v \in V$, $H_u \cap H_v \cap I_{uv} \neq \emptyset$ where the interval $I_{uv}$ is the union of all $uv$-shortest paths.
Problem

Given a graph $G$ assign a label $L_u$ to each node $u$ s.t. for all $s, t \notin G(s, t)$ can be computed from $L_s$ and $L_t$.

Hub sets

Given a graph $G$, assign a hub set $H_u \subseteq V(G)$ to each node $u$, s.t. for all $u, v$ there exists $a \in H_u \cap H_v$ with $a \in P_{uv}$.

Distance labels: $L_u = \{(a, d(u, a)) : a \in H_u\}$

Distance query: $\text{Dist}(L_s, L_t) = \min_{a \in H_s \cap H_t} d(s, a) + d(a, t)$
in $O(|H_s| + |H_t|)$ time.

Covering property: for all $u, v \in V$, $H_u \cap H_v \cap I_{uv} \neq \emptyset$ where the interval $I_{uv}$ is the union of all $uv$-shortest paths.
Distance labeling [Gavoille, Peleg, Pérennes, Raz '04]

Problem
Given a graph $G$ assign a label $L_u$ to each node $u$ s.t. for all $s, t$ $d_G(s, t)$ can be computed from $L_s$ and $L_t$.

Hub sets
Given a graph $G$, assign a hub set $H_u \subseteq V(G)$ to each node $u$, s.t. for all $u, v$ there exists $a \in H_u \cap H_v$ with $a \in P_{uv}$.

Distance labels: $L_u = \{(a, d(u, a)) : a \in H_u\}$
Distance query: $\text{Dist} (L_s, L_t) = \min_{a \in H_s \cap H_t} d(s, a) + d(a, t)$ in $O(|H_s| + |H_t|)$ time.

Covering property: for all $u, v \in V$, $H_u \cap H_v \cap I_{uv} \neq \emptyset$ where the interval $I_{uv}$ is the union of all $uv$-shortest paths.
Distance labeling [Gavoille, Peleg, Pérennes, Raz ’04]

Problem
Given a graph $G$ assign a label $L_u$ to each node $u$ s.t. for all $s, t \ d_G(s, t)$ can be computed from $L_s$ and $L_t$.

Hub sets
Given a graph $G$, assign a hub set $H_u \subseteq V(G)$ to each node $u$, s.t. for all $u, v$ there exists $a \in H_u \cap H_v$ with $a \in P_{uv}$.

Distance labels : $L_u = \{(a, d(u, a)) : a \in H_u\}$

Distance query : $\text{Dist} (L_s, L_t) = \min_{a \in H_s \cap H_t} d(s, a) + d(a, t)$ in $O(|H_s| + |H_t|)$ time.

Covering property : for all $u, v \in V$, $H_u \cap H_v \cap I_{uv} \neq \emptyset$ where the interval $I_{uv}$ is the union of all $uv$-shortest paths.
Labeling with hub sets: hub labeling a.k.a. 2-hop labeling

**Exercise**: Hub labeling for a path?, a tree? a graph with treewidth $k$? a planar graph?

**Open pb**: Increase the best known lower bound for unweighted planar graphs ($\Omega(n^{1/3})$ [Gavoille et al. 2004]).

**Exercise**: Hub labeling for a grid?
Labeling with hub sets: hub labeling a.k.a. 2-hop labeling

Exercise: Hub labeling for a path?, a tree? a graph with treewidth $k$? a planar graph?

Open pb: Increase the best known lower bound for unweighted planar graphs ($\Omega(n^{1/3}) \; [Gavoille \ et \ al. \ 2004]$).

Exercise: Hub labeling for a grid?
Labeling with hub sets: hub labeling a.k.a. 2-hop labeling

**Exercise**: Hub labeling for a path?, a tree? a graph with treewidth k? a planar graph?

**Open pb**: Increase the best known lower bound for unweighted planar graphs ($\Omega(n^{1/3})$ [Gavoille et al. 2004]).

**Exercise**: Hub labeling for a grid?
Problem: Find hub sets \((H_u)_{u \in V}\) of min. size \(|H| = \sum_u |H_u|\).

Greedy cover all shortest paths:
- smallest avg. hub size is equivalent to min. cost set cover,
- \(O(\log n)\)-approximation is possible:
  - select a hub \(x\) and a subset \(K\) of nodes s.t.
    \[
    \text{rel. cost} \frac{|K|}{\text{nb path cov. if } x \text{ added to all } (H_u)_{u \in K}} \text{ is min.}
    \]
  - add hub \(x\) to \(H_u\) for \(u \in K\).

Problem: set cover instance with \(n \times 2^n\) sets!

Solution: fix \(x\), what is the best \(K\)?
- \(G_x\) graph with edges \(uv\) s.t. \(P_{uv}\) still uncov. and \(a \in P_{uv}\).

Exercise: Propose a greedy algorithm for 2-approximating the best \(K\).

Corollary: Hubsets with smallest average size can be \(O(\log n)\)-approximated in polynomial time.
2-hop labeling [Cohen, Halperin, Kaplan, Zwick '03]

Problem: Find hub sets \((H_u)_{u \in V}\) of min. size \(|H| = \sum_u |H_u|\).

Greedy cover all shortest paths:
- smallest avg. hub size is equivalent to min. cost set cover,
- \(O(\log n)\)-approximation is possible:
  - select a hub \(x\) and a subset \(K\) of nodes s.t.
    \[
    \text{rel. cost} \cdot \frac{|K|}{\text{nb path cov. if } x \text{ added to all } (H_u)_{u \in K}} \text{ is min.}
    \]
  - add hub \(x\) to \(H_u\) for \(u \in K\).

Problem: set cover instance with \(n \times 2^n\) sets!

Solution: fix \(x\), what is the best \(K\)?
- \(G_x\) graph with edges \(uv\) s.t. \(P_{uv}\) still uncov. and \(a \in P_{uv}\).

Exercise: Propose a greedy algorithm for 2-approximating the best \(K\).

Corollary: Hubsets with smallest average size can be \(O(\log n)\)-approximated in polynomial time.
2-hop labeling [Cohen, Halperin, Kaplan, Zwick '03]

Problem: Find hub sets \((H_u)_{u \in V}\) of min. size \(|H| = \sum_u |H_u|\).

Greedy cover all shortest paths:
- smallest avg. hub size is equivalent to min. cost set cover,
- \(O(\log n)\)-approximation is possible:
  - select a hub \(x\) and a subset \(K\) of nodes s.t.
    \[
    \text{rel. cost} \frac{|K|}{\text{nb path cov. if } x \text{ added to all } (H_u)_{u \in K}} \text{ is min.}
    \]
  - add hub \(x\) to \(H_u\) for \(u \in K\).

Problem: set cover instance with \(n \times 2^n\) sets!

Solution: fix \(x\), what is the best \(K\)?
\(G_x\) graph with edges \(uv\) s.t. \(P_{uv}\) still uncov. and \(a \in P_{uv}\).

Exercise: Propose a greedy algorithm for 2-approximating the best \(K\).

Corollary: Hubsets with smallest average size can be \(O(\log n)\)-approximated in polynomial time.
2-hop labeling [Cohen, Halperin, Kaplan, Zwick '03]

Problem: Find hub sets $(H_u)_{u \in V}$ of min. size $|H| = \sum_u |H_u|$.

Greedy cover all shortest paths:
- smallest avg. hub size is equivalent to min. cost set cover,
- $O(\log n)$-approximation is possible:
  - select a hub $x$ and a subset $K$ of nodes s.t.
    - rel. cost $\frac{|K|}{\text{nb path cov. if } x \text{ added to all } (H_u)_{u \in K}}$ is min.
  - add hub $x$ to $H_u$ for $u \in K$.

Problem: set cover instance with $n \times 2^n$ sets!

Solution: fix $x$, what is the best $K$?
  - $G_x$ graph with edges $uv$ s.t. $P_{uv}$ still uncov. and $a \in P_{uv}$.

Exercise: Propose a greedy algorithm for 2-approximating the best $K$.

Corollary: Hubsets with smallest average size can be $O(\log n)$-approximated in polynomial time.
Problem: Find hub sets \((H_u)_{u \in V}\) of min. size \(|H| = \sum_u |H_u|\).

Greedy cover all shortest paths:
- smallest avg. hub size is equivalent to min. cost set cover,
- \(O(\log n)\)-approximation is possible:
  - select a hub \(x\) and a subset \(K\) of nodes s.t.
    - \(\text{rel. cost } \frac{|K|}{\text{nb path cov. if } x \text{ added to all } (H_u)_{u \in K}}\) is min.
  - add hub \(x\) to \(H_u\) for \(u \in K\).

Problem: set cover instance with \(n \times 2^n\) sets!

Solution: fix \(x\), what is the best \(K\)?
- \(G_x\) graph with edges \(uv\) s.t. \(P_{uv}\) still uncov. and \(a \in P_{uv}\).

Exercise: Propose a greedy algorithm for 2-approximating the best \(K\).

Corollary: Hubsets with smallest average size can be \(O(\log n)\)-approximated in polynomial time.
Problem: Find hub sets \((H_u)_{u \in V}\) of min. size \(|H| = \sum_u |H_u|\).

Greedy cover all shortest paths:
- smallest avg. hub size is equivalent to min. cost set cover,
- \(O(\log n)\)-approximation is possible:
  - select a hub \(x\) and a subset \(K\) of nodes s.t.
    \[\text{rel. cost} \frac{|K|}{\text{nb path cov. if } x \text{ added to all } (H_u)_{u \in K}}\]
    is min.
- add hub \(x\) to \(H_u\) for \(u \in K\).

Problem: set cover instance with \(n \times 2^n\) sets!

Solution: fix \(x\), what is the best \(K\)?
\(G_x\) graph with edges \(uv\) s.t. \(P_{uv}\) still uncov. and \(a \in P_{uv}\).

Exercise: Propose a greedy algorithm for 2-approximating the best \(K\).

Corollary: Hubsets with smallest average size can be \(O(\log n)\)-approximated in polynomial time.
Highway dimension [Abraham, Delling, Fiat, Goldberg, Werneck ’10-13]

**Definition**

Highway dimension \( h = \max_{u,r} \min_H \text{hitting set of } P_{ur} |H| \)

where \( P_{ur} = \{ P \in P_r \mid \bar{P} \cap B(u, r) \neq \emptyset \} \), \( P_r = \{ P \mid \ell(P) > \frac{r}{2} \} \),

and \( \bar{P} \) is any shortest path extending \( P \) by 0 or 1 edge at each extremity.

**Theorem**

Any graph \( G \) with highway dimension \( h \) and diameter \( D \) admits a node ordering \( \pi \) s.t. \( CH^{opt}_\pi \) produces at most \( O(nh \log D) \) shortcuts and \( CH_\pi + \text{RP bidir. Dij. visits} \) \( O(h \log D) \) nodes.

**Exercise**: use \( CH \) to compute a HHL.
Highway dimension [Abraham, Delling, Fiat, Goldberg, Werneck ’10-13]

**Definition**

Highway dimension $h = \max_{u,r} \min_{H} |H|$ hitting set of $\mathcal{P}_{ur}$ where $\mathcal{P}_{ur} = \{P \in \mathcal{P}_r \mid \bar{P} \cap B(u, r) \neq \emptyset\}$, $\mathcal{P}_r = \{P \mid \ell(P) > \frac{r}{2}\}$, and $\bar{P}$ is any shortest path extending $P$ by 0 or 1 edge at each extremity.

**Theorem**

Any graph $G$ with highway dimension $h$ and diameter $D$ admits a node ordering $\pi$ s.t. $CH_{\pi}^{opt}$ produces at most $O(nh \log D)$ shortcuts and $CH_{\pi} + RP$ bidir. Dij. visits $O(h \log D)$ nodes.

**Exercise**: use $CH$ to compute a HHL.
Highway dimension [Abraham, Delling, Fiat, Goldberg, Werneck '10-13]

Definition
Highway dimension \( h = \max_{u,r} \min_H |H| \) hitting set of \( P_{ur} \)
where \( P_{ur} = \{ P \in P_r \mid \bar{P} \cap B(u, r) \neq \emptyset \} \), \( P_r = \{ P \mid \ell(P) > \frac{r}{2} \} \),
and \( \bar{P} \) is any shortest path extending \( P \) by 0 or 1 edge at each extremity.

Theorem
Any graph \( G \) with highway dimension \( h \) and diameter \( D \) admits a node ordering \( \pi \) s.t. \( \text{CH}^\text{opt}_\pi \) produces at most \( O(nh \log D) \) shortcuts and \( \text{CH}_\pi + \text{RP bidir. Di} \) visits \( O(h \log D) \) nodes.

Exercise: use \( \text{CH} \) to compute a HHL.
Hierarchical Hub Labeling (HHL) [BGKSW'15]
A hub labeling is hierarchical if it respects an order \( \pi \) such that hubs are more important: \( v \in H_u \Rightarrow v \geq_{\pi} u \) (the graph with edges from nodes to their hubs is a DAG).

Canonical HHL
Given an ordering \( \pi \), for all \( u, v \) add \( \max_{\pi} I_{uv} \) to \( H_u \) and \( H_v \).

Proposition
Canonical HHL for \( \pi \) is the minimum HHL that respects \( \pi \).

Exercise: show that any minimal HHL is canonical.
Hierarchical Hub Labeling (HHL) [BGKSW’15]
A hub labeling is hierarchical if it respects an order $\pi$ such that hubs are more important: $v \in H_u \Rightarrow v \geq_{\pi} u$
(the graph with edges from nodes to their hubs is a DAG).

Canonical HHL
Given an ordering $\pi$, for all $u, v$ add $\max_{\pi} I_{uv}$ to $H_u$ and $H_v$.

Proposition
Canonical HHL for $\pi$ is the minimum HHL that respects $\pi$.

Exercise: show that any minimal HHL is canonical.
Hierarchical Hub Labeling (HHL) [BGKSW’15]
A hub labeling is hierarchical if it respects an order $\pi$ such that hubs are more important: $v \in H_u \Rightarrow v \geq_\pi u$ (the graph with edges from nodes to their hubs is a DAG).

Canonical HHL
Given an ordering $\pi$, for all $u, v$ add $\max_\pi I_{uv}$ to $H_u$ and $H_v$.

Proposition
Canonical HHL for $\pi$ is the minimum HHL that respects $\pi$.

Exercise: show that any minimal HHL is canonical.
Hierarchical Hub Labeling (HHL) [BGKSW'15]
A hub labeling is hierarchical if it respects an order $\pi$ such that hubs are more important: $v \in H_u \Rightarrow v \geq_\pi u$
(the graph with edges from nodes to their hubs is a DAG).

Canonical HHL
Given an ordering $\pi$, for all $u, v$ add $\max_\pi I_{uv}$ to $H_u$ and $H_v$.

Proposition
Canonical HHL for $\pi$ is the minimum HHL that respects $\pi$.

Exercise: show that any minimal HHL is canonical.
Pruned Labeling [Akiba, Iwata, Yoshida ’13]

Procedure \textbf{PrunedLab} \((G, \pi)\)
- Distance labels \(L_u := \emptyset\) for all \(u\).
- \textbf{For each} \(a \in V(G)\) \textbf{in decreasing order of} \(\pi\) \textbf{do}
  - \textbf{PrunedDijkstra} \((G, a, L)\)
  - Add \((a, d(a, u))\) to \(L_u\) for each visited node \(u\).

Procedure \textbf{PrunedDijkstra} \((G, a, L)\)
- Starting from \(u = a\), visit \(u\) if \(d(u) < \text{Dist}(L_a, L_u)\).

\textbf{Theorem}
PL computes the canonical HHL associated to \(\pi\) in \(O(nL \log n + mL^2)\) time where \(L\) is maximum label size.

Open pb: charac. classes of graphs with \(|HHL| = O(|HL|)\).
Pruned Labeling [Akiba, Iwata, Yoshida ’13]

Procedure **PrunedLab** \((G, \pi)\)
- Distance labels \(L_u := \emptyset\) for all \(u\).
- For each \(a \in V(G)\) in decreasing order of \(\pi\) do
  - **PrunedDijkstra** \((G, a, L)\)
    - Add \((a, d(a, u))\) to \(L_u\) for each visited node \(u\).

Procedure **PrunedDijkstra** \((G, a, L)\)
- Starting from \(u = a\), visit \(u\) if \(d(u) < \text{Dist}(L_a, L_u)\).

**Theorem**

PL computes the canonical HHL associated to \(\pi\) in \(O(nL \log n + mL^2)\) time where \(L\) is maximum label size.

**Open pb**: charac. classes of graphs with \(|HHL| = O(|HL|)\).
HL on massive networks [Delling, Goldberg, Pajor, Werneck ’14]

HHL using **random sampling** to approximate greedy cover (for $\pi$) in combination with **pruned labeling** (for hub sets).

$O(\log n)$ approximation in theory, smallest hub labelings in practice (and fastest distance oracle).
HL on massive networks [Delling, Goldberg, Pajor, Werneck ’14]

HHL using random sampling to approximate greedy cover (for $\pi$) in combination with pruned labeling (for hub sets).

$O(\log n)$ approximation in theory, smallest hub labelings in practice (and fastest distance oracle).
**Skeleton dimension [Kosowski et al. ’17]**

Graph property ensuring small hub sets.

The skeleton dimension $k$ of $G$ is the maximum “width” of a “pruned” shortest path tree (see pres.).

**Theorem**

Any graph $G$ with skeleton dimension $k$ and diameter $D$ has hub sets of size $O(k \log \log k \log D)$ (polyn. time constr. w.h.p.).

Open pb: what additional property ensures efficient Reach/CH?

Open pb: tight bounds on HL/HHL in grids?
**Skeleton dimension** [Kosowski et al. ’17]

**Graph property** ensuring small hub sets.

The **skeleton dimension** $k$ of $G$ is the maximum “width” of a “pruned” shortest path tree (see pres.).

**Theorem**
Any graph $G$ with skeleton dimension $k$ and diameter $D$ has hub sets of size $O(k \log \log k \log D)$ (polyn. time constr. w.h.p.).

**Open pb**: what additional property ensures efficient Reach/CH?

**Open pb**: tight bounds on HL/HHL in grids?
Skeleton dimension [Kosowski et al. ’17]

**Graph property** ensuring small hub sets.

The **skeleton dimension** $k$ of $G$ is the maximum “width” of a “pruned” shortest path tree (see pres.).

**Theorem**
Any graph $G$ with skeleton dimension $k$ and diameter $D$ has hub sets of size $O(k \log \log k \log D)$ (polyn. time constr. w.h.p.).

**Open pb:** what additional property ensures efficient Reach/CH?

**Open pb:** tight bounds on HL/HHL in grids?
Skeleton dimension [Kosowski et al. ’17]

Graph property ensuring small hub sets.

The skeleton dimension $k$ of $G$ is the maximum “width” of a “pruned” shortest path tree (see pres.).

**Theorem**

Any graph $G$ with skeleton dimension $k$ and diameter $D$ has hub sets of size $O(k \log \log k \log D)$ (polyn. time constr. w.h.p.).

**Open pb**: what additional property ensures efficient Reach/CH?

**Open pb**: tight bounds on HL/HHL in grids?
Skeleton dimension [Kosowski et al. ’17]

Graph property ensuring small hub sets.

The skeleton dimension $k$ of $G$ is the maximum “width” of a “pruned” shortest path tree (see pres.).

**Theorem**

Any graph $G$ with skeleton dimension $k$ and diameter $D$ has hub sets of size $O(k \log \log k \log D)$ (polyn. time constr. w.h.p.).

**Open pb**: what additional property ensures efficient Reach/CH?

**Open pb**: tight bounds on HL/HHL in grids?
Barcelona shortest path tree
Barcelona skeleton: prune last third
Barcelona skeleton: prune last third
Barcelona skeleton: prune last third
Barcelona skeleton: prune last third
Barcelona skeleton: prune last third
Barcelona skeleton : prune last third
Barcelona skeleton: prune last third
Barcelona skeleton: prune last third
Barcelona skeleton: prune last third
Barcelona skeleton: prune last third
Barcelona skeleton: prune last third
Tree skeleton

\[ P_{uv} \]

\[ u \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \right}
Tree skeleton

\[ \begin{align*}
P_{uv} & \quad 2/3 \\
\{ w \in P_{uv} : d(u,w) \leq \frac{2}{3} d(u,v) \} \\
1/3
\end{align*} \]
Tree skeleton

\[ \text{Reach}_{P_{uv}}(w) \geq \frac{d(u, v)}{2} \]
Tree skeleton

\[ T_u = \bigcup_P P_{uv} \]
Tree skeleton

\[ T_u^* = \bigcup_v P_{uv}^{2/3} \]

\[ T_u = \bigcup_v P_{uv} \]
Tree skeleton

\[ T_u^* = \bigcup_v P_{uv}^{2/3} \]

\[ T_u = \bigcup_v P_{uv} \]

\[ \text{Cut}_r(T_u) \]
Tree skeleton

$T_u^* = \bigcup_v P_{uv}^{2/3}$

$T_u = \bigcup_v P_{uv}$

$\text{Width}(T_u^*) = \max_{\ell} |\text{Cut}_\ell(T_u^*)|$
Tree skeleton

\[ T_u^* = \bigcup_{v} P_{uv}^{2/3} \]

\[ T_u = \bigcup_{v} P_{uv} \]

skel. dim. \quad k = \max_u \text{Width}(T_u^*)
Hub set selection
Hub set selection
Hub set selection

\[ \omega \text{ s.t. } \rho(\omega) \min \text{ in } P_{uv}^{2/3} \cap P_{vu}^{2/3} \]
Hub set selection
Hub set selection

\[ \frac{d(u, w)}{6} \quad \frac{d(u, y)}{6} \quad \omega = x_y \quad \text{s.t.} \]

\[ \frac{1}{3} \]

\[ \frac{2}{3} \]

\[ T_u \]

\[ T_v \]

\[ u \]

\[ v \]
Hub set selection

\[ u \quad T_u^* \]

\[ \frac{d(u, w)}{6} \]

\[ \frac{d(u, y)}{6} \]

\[ x_w \]

\[ \omega \text{ s.t. } p(\omega) \min \text{ in } P_{xw} \alpha \text{ in } P_{wy} \]

\[ 2/3 \]

\[ T_v^* \]
Hub set selection (proof for unweighted graphs)

Draw $\rho(w) \in [0, 1]$ u.a.r. for all $w \in V(G)$.

$$H_u = \{ w \mid \rho(w) \text{ min. in } P_{xw}, \} \cup \{ x_y \mid \rho(x_y) \text{ min. in } P_{xyy} \}$$

(Can be computed in $\tilde{O}(n + m)$ separately for each node with shared randomness.)

A sub-path $P_{xy}$ has length $\frac{d(u, y)}{6}$ and generates a hub in $H_u$ with probability at most $\frac{12}{d(u, y)}$.

$$E[|H_u|] \leq \sum_{y \in V(T_u^*)} \frac{12}{d(u, y)} \leq \sum_r |\text{Cut}_r(T_u^*)| \frac{12}{r} = O(k \log D)$$
Hub set selection (proof for unweighted graphs)

Draw $\rho(w) \in [0, 1]$ u.a.r. for all $w \in V(G)$.

$H_u = \{ w \mid \rho(w) \text{ min. in } P_{xw} \} \cup \{ xy \mid \rho(xy) \text{ min. in } P_{xy} \}$

(Can be computed in $\tilde{O}(n + m)$ separately for each node with shared randomness.)

A sub-path $P_{xy}$ has length $\frac{d(u,y)}{6}$ and generates a hub in $H_u$ with probability at most $\frac{12}{d(u,y)}$.

$$E[|H_u|] \leq \sum_{y \in V(T_u^*)} \frac{12}{d(u,y)} \leq \sum_r |\text{Cut}_r(T_u^*)| \frac{12}{r} = O(k \log D)$$
Hub set selection (proof for unweighted graphs)

Draw $\rho(w) \in [0, 1]$ u.a.r. for all $w \in V(G)$.

$$H_u = \{w \mid \rho(w) \text{ min. in } P_{xww}\} \cup \{x_y \mid \rho(x_y) \text{ min. in } P_{xyy}\}$$

(Can be computed in $\tilde{O}(n + m)$ separately for each node with shared randomness.)

A sub-path $P_{xyy}$ has length $\frac{d(u, y)}{6}$ and generates a hub in $H_u$ with probability at most $\frac{12}{d(u, y)}$.

$$E[|H_u|] \leq \sum_{y \in V(T_u^*)} \frac{12}{d(u, y)} \leq \sum r |\text{Cut}_r(T_u^*)| \frac{12}{r} = O(k \log D)$$
Road networks: two tree skeletons
What ...maps do?
What ... maps do?
What ...maps do?
**Dimension of grids**

\[
h = \Theta(\sqrt{n})
\]

\[
k = \Theta(\log n)
\]
Dimension of grids

\[ h = \Theta(\sqrt{n}) \]

\[ k = \Theta(\log n) \]
Dimension of grids

\[ h = \Theta(\sqrt{n}) \]

\[ k = \Theta(\log n) \]
Dimension of grids

\[ h = \Theta(\sqrt{n}) \]

\[ k = \Theta(\log n) \]
Highway vs skeleton in Brooklyn

Packing of 172 paths

Skeleton width 48
Open: random grid (here $500 \times 500$)

$k = 70$

$k = 49$ (fpp $[1,4]$)

$k = 49$ (prob $2/3$)
What about general graphs?
Pre-hub labeling [Angelidakis, Makarychev, Oparin '17]

Hub sets $(H_u)_{u \in V(G)}$ for a graph $G$ form a pre-hub labeling if for all $u, v$ pairs, hubs cross on $P_{uv}$: $\exists u' \in P_{uv} \cap H_u$ and $\exists v' \in P_{uv} \cap H_v$ with $u' \in P_{v'v}$ and $v' \in P_{uu'}$.

**Theorem**
If shortest paths are unique,
- PHL 2-approximate HL (and pol. time constr.),
- PHL can be converted to HL with $O(\log D)$ factor.

**Theorem**
If shortest paths are not unique,
- best polyn. time approx. is $\Omega(\log n)$ (even if $D = O(1)$).

**Theorem**
In trees, HHL 2-approximate HL (and pol. time constr.).

**Exercise**: prove it.
Pre-hub labeling [Angelidakis, Makarychev, Oparin '17]

Hub sets \((H_u)_{u \in V(G)}\) for a graph \(G\) form a pre-hub labeling if for all \(u, v\) pairs, hubs cross on \(P_{uv}\): \(\exists u' \in P_{uv} \cap H_u\) and \(\exists v' \in P_{uv} \cap H_v\) with \(u' \in P_{v'v}\) and \(v' \in P_{uu'}\).

**Theorem**
If shortest paths are unique,
- PHL 2-approximate HL (and pol. time constr.),
- PHL can be converted to HL with \(O(\log D)\) factor.

**Theorem**
If shortest paths are not unique,
- best polyn. time approx. is \(\Omega(\log n)\) (even if \(D = O(1)\)).

**Theorem**
In trees, HHL 2-approximate HL (and pol. time constr.).

**Exercise**: prove it.
Pre-hub labeling [Angelidakis, Makarychev, Oparin ’17]

Hub sets \((H_u)_{u \in V(G)}\) for a graph \(G\) form a pre-hub labeling if for all \(u, v\) pairs, hubs cross on \(P_{uv}\): \(\exists u' \in P_{uv} \cap H_u\) and \(\exists v' \in P_{uv} \cap H_v\) with \(u' \in P_{v'v}\) and \(v' \in P_{uu'}\).

**Theorem**

If shortest paths are unique,
- PHL 2-approximate HL (and pol. time constr.),
- PHL can be converted to HL with \(O(\log D)\) factor.

**Theorem**

If shortest paths are not unique,
- best polyn. time approx. is \(\Omega(\log n)\) (even if \(D = O(1)\)).

**Theorem**

In trees, HHL 2-approximate HL (and pol. time constr.).

**Exercise**: prove it.
Pre-hub labeling [Angelidakis, Makarychev, Oparin ’17]

Hub sets \((H_u)_{u \in V(G)}\) for a graph \(G\) form a \textit{pre-hub labeling} if for all \(u, v\) pairs, hubs cross on \(P_{uv}\): \(\exists u' \in P_{uv} \cap H_u\) and \(\exists v' \in P_{uv} \cap H_v\) with \(u' \in P_{v'v}\) and \(v' \in P_{uu'}\).

**Theorem**

If shortest paths are unique,
- PHL 2-approximate HL (and pol. time constr.),
- PHL can be converted to HL with \(O(\log D)\) factor.

**Theorem**

If shortest paths are \textit{not} unique,
- best polyn. time approx. is \(\Omega(\log n)\) (even if \(D = O(1)\)).

**Theorem**

In trees, HHL 2-approximate HL (and pol. time constr.).

**Exercise**: prove it.
Pre-hub labeling [Angelidakis, Makarychev, Oparin '17]

Hub sets \((H_u)_{u \in V(G)}\) for a graph \(G\) form a pre-hub labeling if for all \(u, v\) pairs, hubs cross on \(P_{uv}\): \(\exists u' \in P_{uv} \cap H_u\) and \(\exists v' \in P_{uv} \cap H_v\) with \(u' \in P_{v'v}\) and \(v' \in P_{uu'}\).

**Theorem**
If shortest paths are unique,
- PHL 2-approximate HL (and pol. time constr.),
- PHL can be converted to HL with \(O(\log D)\) factor.

**Theorem**
If shortest paths are **not** unique,
- best polyn. time approx. is \(\Omega(\log n)\) (even if \(D = O(1)\)).

**Theorem**
In trees, HHL 2-approximate HL (and pol. time constr.).

**Exercise**: prove it.
Hub labeling of general graphs

Any graph has $O\left(\frac{n}{\log n}\right)$ hubsets for $m = O(n)$ (combining [Kosowski et al. '17] and [Alstrup et al. '16]).

Idea (for $\Delta = O(1)$):
- for $r \geq \delta$, the width of a $r$-cut of a skeleton tree is $k = O(n/\delta)$ (we can get $O(n/\delta)$ hubsets for distances $\geq \delta$),
- select as hubs all nodes at distance less than $\delta = \frac{\log n}{2\log \Delta}$ (at most $\sqrt{n}$ nodes).
Hub labeling of general graphs

Any graph has $O\left(\frac{n}{\log n}\right)$ hubsets for $m = O(n)$ (combining [Kosowski et al. ’17] and [Alstrup et al. ’16]).

Idea (for $\Delta = O(1)$):
- for $r \geq \delta$, the width of a $r$-cut of a skeleton tree is $k = O(n/\delta)$ (we can get $O(n/\delta)$ hubsets for distances $\geq \delta$),
- select as hubs all nodes at distance less than $\delta = \frac{\log n}{2\log \Delta}$ (at most $\sqrt{n}$ nodes).
Lower bound on hub labeling of general graphs

**Theorem** [Kosowski et al. ‘19]: There exists graphs with max-degree 3 such that any hubsets have average size $\Omega\left(\frac{n}{2^{O(\sqrt{\log n})}}\right)$.

Linked to Ruzsa-Szemerédi function bounding the number of edges in a graph decomposable into $n$ induced matchings.

There exists graphs with max-degree 3 such that any distance labels must have average size $\geq \text{SUMINDEX}(n/O(\sqrt{\log n}))$.

**Open problem**: does any sparse graph have a (centralized) distance oracle of size $O(n^{1.5})$ and query time $O(n^{0.5})$?
Lower bound on hub labeling of general graphs

**Theorem** [Kosowski et al. ‘19]: There exists graphs with max-degree 3 such that any hubsets have average size $\Omega\left(\frac{n}{2^{O(\sqrt{\log n})}}\right)$.

Linked to Ruzsa-Szemerédi function bounding the number of edges in a graph decomposable into $n$ induced matchings.

There exists graphs with max-degree 3 such that any distance labels must have average size $\geq \text{SUMINDEX}(n/O(\sqrt{\log n}))$.

**Open problem**: does any sparse graph have a (centralized) distance oracle of size $O(n^{1.5})$ and query time $O(n^{0.5})$?
Lower bound on hub labeling of general graphs

Theorem [Kosowski et al. ’19]: There exists graphs with max-degree 3 such that any hubsets have average size \( \Omega\left(\frac{n}{2^{O(\sqrt{\log n})}}\right) \).

Linked to Ruzsa-Szemerédi function bounding the number of edges in a graph decomposable into \( n \) induced matchings.

There exists graphs with max-degree 3 such that any distance labels must have average size

\[ \geq \text{SUMINDEX}(n/O(\sqrt{\log n})) \]

Open problem: does any sparse graph have a (centralized) distance oracle of size \( O(n^{1.5}) \) and query time \( O(n^{0.5}) \)?
Lower bound on hub labeling of general graphs

**Theorem** [Kosowski et al. ’19]: There exists graphs with max-degree 3 such that any hubsets have average size $\Omega\left(\frac{n}{2^{O(\sqrt{\log n})}}\right)$.

Linked to Ruzsa-Szemerédi function bounding the number of edges in a graph decomposable into $n$ induced matchings.

There exists graphs with max-degree 3 such that any distance labels must have average size

$$\geq \text{SUMINDEX}(n/O(\sqrt{\log n})).$$

**Open problem:** does any sparse graph have a (centralized) distance oracle of size $O(n^{1.5})$ and query time $O(n^{0.5})$?
Each $V_i$ is a regular $2^\ell \times \cdots \times 2^\ell$ lattice of dim. $\ell \approx \sqrt{\log n}$ (here $\ell = 2$). Edges from $V_{i-1}$ to $V_i$ connect nodes differing on $i$th coordinate.
A graph is an RS-graph if it can be decomposed into $n$ induced matchings.
A graph is an RS-graph if it can be decomposed into $n$ induced matchings.
A graph is an **RS-graph** if it can be decomposed into $n$ induced matchings.
A graph is an **RS-graph** if it can be decomposed into $n$ induced matchings.
A graph is an **RS-graph** if it can be decomposed into \( n \) induced matchings.

What are the densest RS-graphs?

**Theorem ([Ruzsa, Szemerédi ’78])**

Any RS-graph has at most \( \frac{n^2}{2^{O(\log^* n)}} \) edges.

Define \( RS(n) \) as the smallest integer such that there exists an RS-graph with \( n \) nodes and \( \frac{n^2}{RS(n)} \) edges.

\[
2^{\Omega(\log^* n)} \leq RS(n) \leq 2^{O(\sqrt{\log n})}
\]

[Ruzsa, Szemerédi ’78] [Elkin ’10] [Fox ’11]
A graph is an **RS-graph** if it can be decomposed into $n$ induced matchings.

What are the **densest RS-graphs**?

**Theorem ([Ruzsa, Szemerédi ‘78])**
Any RS-graph has at most $\frac{n^2}{2^{O(\log^* n)}}$ edges.

Define $RS(n)$ as the smallest integer such that there exists an RS-graph with $n$ nodes and $\frac{n^2}{RS(n)}$ edges.

$$2^{\Omega(\log^* n)} \leq RS(n) \leq 2^{O(\sqrt{\log n})}$$

[Ruzsa, Szemerédi ‘78] [Elkin ‘10] [Fox ‘11]
A graph is an RS-graph if it can be decomposed into $n$ induced matchings.

What are the densest RS-graphs?

**Theorem ([Ruzsa, Szemerédi ‘78])**
Any RS-graph has at most $\frac{n^2}{2^{O(\log^* n)}}$ edges.

Define $RS(n)$ as the smallest integer such that there exists an RS-graph with $n$ nodes and $\frac{n^2}{RS(n)}$ edges.

$$2^{\Omega(\log^* n)} \leq RS(n) \leq 2^{O(\sqrt{\log n})}$$

[Ruzsa, Szemerédi ‘78] [Elkin ‘10] [Fox ‘11]
A graph is an **RS-graph** if it can be decomposed into $n$ induced matchings.

What are the **densest RS-graphs**?

**Theorem ([Ruzsa, Szemerédi ‘78])**
Any RS-graph has at most $\frac{n^2}{2^{O(\log^* n)}}$ edges.

Define $RS(n)$ as the smallest integer such that there exists an RS-graph with $n$ nodes and $\frac{n^2}{RS(n)}$ edges.

$$2^{\Omega(\log^* n)} \leq RS(n) \leq 2^{O(\sqrt{\log n})}$$

[Ruzsa, Szemerédi ‘78] [Elkin ‘10] [Fox ‘11]
A graph is an **RS-graph** if it can be decomposed into \( n \) induced matchings.

What are the **densest RS-graphs**?

**Theorem** ([Ruzsa, Szemerédi ’78])
Any **RS-graph** has at most \( \frac{n^2}{2^\Omega(\log^* n)} \) edges.

Define **RS**\((n)\) as the smallest integer such that there exists an **RS-graph** with \( n \) nodes and \( \frac{n^2}{\text{RS}(n)} \) edges.

\[ 2^{\Omega(\log^* n)} \leq \text{RS}(n) \leq 2^{O(\sqrt{\log n})} \]

[Ruzsa, Szemerédi ’78] [Elkin ’10] [Fox ’11]
\( G^D_y = \{ x_0 z_{2\ell} \mid y = \frac{x+z}{2} \text{ and } d_G(x, z) = D \} \quad \exists D \text{ s.t. } |\bigcup_y G^D_y| \geq \frac{n^2}{2^{O(\sqrt{\log n})}} \)
Connection with SumIndex

\[ \text{SUMINDEX}(n) = \min_{\text{Encoder}} \max_X |M_A| + |M_B| \]

\[ \Omega(\sqrt{n}) \leq \text{SUMINDEX}(n) \leq \tilde{O}\left(\frac{n}{2^{\sqrt{\log n}}}\right) \]

[Pudlak 1994] [Babai et al, 2003] [Ambainis 1996]
Connection with SumIndex

\[
\text{SUMINDEX}(n) = \min_{\text{Encoder}} \max_{X} |M_A| + |M_B|
\]

\[
\Omega(\sqrt{n}) \leq \text{SUMINDEX}(n) \leq \tilde{O}\left(\frac{n}{2\sqrt{\log n}}\right)
\]

[Pudlak 1994] [Babai et al, 2003] [Ambainis 1996]
\[ G_X = G \setminus \{ y_\ell \mid X_y = 0 \} , \text{ send } x = 2a, L_{x_0}, z = 2b, L_{z_{2\ell}} , \text{ check } d(x_0, z_{2\ell}). \]

\[ \text{SUMINDEX}\left( \frac{n}{2^{O(\ell)}} \right) \leq \text{DistLab}(n) \]
$G_X = G \setminus \{y_\ell \mid X_y = 0\}$, send $x = 2a, L_{x_0}, z = 2b, L_{z_{2\ell}}$, check $d(x_0, z_{2\ell})$.

$\text{SUMINDEX}(n/2^{O(\ell)}) \leq \text{DistLab}(n)$
What about more hops?
h-hop distance

\[ d^h_G(u, v) = \min_{P \text{ uv-path of } \leq h \text{ edges}} \ell(P) \]

Usual distance: \( d_G(u, v) = d^{n-1}_G(u, v) \)
h-hop distance

\[ d_h^G(u, v) = \min_{P \text{ uv-path of } \leq_h \text{ edges}} \ell(P) \]

Usual distance: \( d_G(u, v) = d_G^{n-1}(u, v) \)
Exercise

We define a h-hopset of G as a set H of edges such that $d^h_{G \cup H}(u, v) = d_G(u, v)$ where each edge uv of H is considered to have length $d_G(u, v)$.

(a) What is the minimum number of edges in $G \cup H$ when H is a 1-hopset of G?

(b) What notion seen in course is tightly related to the notion of 2-hopset?

(c) Suppose that G is a path of length n, propose a 3-hopset of G with as few edges as you can (we do not care about multiplicative constants).

(d) Same question for a 4-hopset.

(e) Consider a 3-hopset H of a graph G. Propose a distance oracle based on distinguishing middle edges of 3-hop shortest paths from the two others. What query time do you obtain when G is a path of length n?
Exercise

We define a h-hopset of $G$ as a set $H$ of edges such that $d^h_{G \cup H}(u, v) = d_G(u, v)$ where each edge $uv$ of $H$ is considered to have length $d_G(u, v)$.

(a) What is the minimum number of edges in $G \cup H$ when $H$ is a 1-hopset of $G$?

(b) What notion seen in course is tightly related to the notion of 2-hopset?

(c) Suppose that $G$ is a path of length $n$, propose a 3-hopset of $G$ with as few edges as you can (we do not care about multiplicative constants).

(d) Same question for a 4-hopset.

(e) Consider a 3-hopset $H$ of a graph $G$. Propose a distance oracle based on distinguishing middle edges of 3-hop shortest paths from the two others. What query time do you obtain when $G$ is a path of length $n$?
Exercise

We define a h-hopset of $G$ as a set $H$ of edges such that $d^H_{G\cup H}(u, v) = d_G(u, v)$ where each edge $uv$ of $H$ is considered to have length $d_G(u, v)$.

(a) What is the minimum number of edges in $G \cup H$ when $H$ is a 1-hopset of $G$?

(b) What notion seen in course is tightly related to the notion of 2-hopset?

(c) Suppose that $G$ is a path of length $n$, propose a 3-hopset of $G$ with as few edges as you can (we do not care about multiplicative constants).

(d) Same question for a 4-hopset.

(e) Consider a 3-hopset $H$ of a graph $G$. Propose a distance oracle based on distinguishing middle edges of 3-hop shortest paths from the two others. What query time do you obtain when $G$ is a path of length $n$?
Exercise

We define a **h-hopset of G** as a set $H$ of edges such that $d^h_{G \cup H}(u, v) = d_G(u, v)$ where each edge $uv$ of $H$ is considered to have length $d_G(u, v)$.

(a) What is the minimum number of edges in $G \cup H$ when $H$ is a 1-hopset of $G$?

(b) What notion seen in course is tightly related to the notion of 2-hopset?

(c) Suppose that $G$ is a path of length $n$, propose a 3-hopset of $G$ with as few edges as you can (we do not care about multiplicative constants).

(d) Same question for a 4-hopset.

(e) Consider a 3-hopset $H$ of a graph $G$. Propose a distance oracle based on distinguishing middle edges of 3-hop shortest paths from the two others. What query time do you obtain when $G$ is a path of length $n$?
Exercise

We define a h-hopset of $G$ as a set $H$ of edges such that $d^h_{G\cup H}(u, v) = d_G(u, v)$ where each edge $uv$ of $H$ is considered to have length $d_G(u, v)$.

(a) What is the minimum number of edges in $G\cup H$ when $H$ is a 1-hopset of $G$?

(b) What notion seen in course is tightly related to the notion of 2-hopset?

(c) Suppose that $G$ is a path of length $n$, propose a 3-hopset of $G$ with as few edges as you can (we do not care about multiplicative constants).

(d) Same question for a 4-hopset.

(e) Consider a 3-hopset $H$ of a graph $G$. Propose a distance oracle based on distinguishing middle edges of 3-hop shortest paths from the two others. What query time do you obtain when $G$ is a path of length $n$?
Exercise

We define a h-hopset of $G$ as a set $H$ of edges such that $d^h_{G∪H}(u, v) = d_G(u, v)$ where each edge $uv$ of $H$ is considered to have length $d_G(u, v)$.

(a) What is the minimum number of edges in $G∪H$ when $H$ is a 1-hopset of $G$?

(b) What notion seen in course is tightly related to the notion of 2-hopset?

(c) Suppose that $G$ is a path of length $n$, propose a 3-hopset of $G$ with as few edges as you can (we do not care about multiplicative constants).

(d) Same question for a 4-hopset.

(e) Consider a 3-hopset $H$ of a graph $G$. Propose a distance oracle based on distinguishing middle edges of 3-hop shortest paths from the two others. What query time do you obtain when $G$ is a path of length $n$?
3-Hopsets in Graphs with Bounded Skeleton Dimension

Theorem (Gupta et al. 2019)
For a unique shortest path graph with skeleton dimension $k$ and polylog average link length, there exists a randomized construction of a $3$-hopset distance oracle of size $|H| = O(nk \log k \log \log n)$, which for an arbitrary queried node pair performs distance queries in expected time $O(k^2 \log^2 k \log^2 \log n)$.

Open pb: Does there exists $\varepsilon, \varepsilon' > 0$ and distance oracles for constant degree graphs with size $O(n^{2-\varepsilon})$ and query time $O(n^{1-\varepsilon'})$?

Open pb: Could it be a $3$-hopset distance oracle?
Theorem (Gupta et al. 2019)
For a unique shortest path graph with skeleton dimension \( k \) and polylog average link length, there exists a randomized construction of a 3-hopset distance oracle of size 
\[ |H| = O(n k \log k \log \log n), \]
which for an arbitrary queried node pair performs distance queries in expected time 
\[ O(k^2 \log^2 k \log \log n). \]

Open pb : Does there exists \( \varepsilon, \varepsilon' > 0 \) and distance oracles for constant degree graphs with size \( O(n^{2-\varepsilon}) \) and query time \( O(n^{1-\varepsilon'}) \)?

Open pb : Could it be a 3-hopset distance oracle?
Further reading

[Angelidakis, Makarychev, Oparin 2017]
Algorithmic and hardness results for the hub labeling problem.

[Hatami, Hatami 2022]
The Implicit Graph Conjecture is False.

Open: practical adjacency labeling schemes.
Further reading


Open: practical adjacency labeling schemes.
Further reading


Open: practical adjacency labeling schemes.
Exercise 3 : answer to either A or B

A/ We define a hierarchical hub labeling HHL on a tree $T$ using a centroid $c$ as most important node before recursing on subtrees of $c$ (removing $c$ disconnects $T$ into subtrees of size $\leq n/2$).
Prove that it provides a 2 approximation of the smallest possible hub labeling $HL$ of $T$.
Hint : associate a matching to the centroid and relate it to some node-hub relations of $HL$.

B/ Construct a family of graphs $G_n$ with $n$ nodes such that $|HL_n| = O(|HHL_n|/n^\varepsilon)$ for some $\varepsilon > 0$ where $HL_n$ (resp. $HHL_n$) denotes the size of the smallest hub labeling (resp. smallest hierarchical hub labeling) of $G_n$. 
Exercise 3 : answer to either A or B

A/ We define a hierarchical hub labeling HHL on a tree T using a centroid c as most important node before recursing on subtrees of c (removing c disconnects T into subtrees of size $\leq n/2$).

Prove that it provides a 2 approximation of the smallest possible hub labeling HL of T.

Hint: associate a matching to the centroid and relate it to some node-hub relations of HL.

B/ Construct a family of graphs $G_n$ with n nodes such that $|HL_n| = O(|HHL_n|/n^\varepsilon)$ for some $\varepsilon > 0$ where $HL_n$ (resp. $HHL_n$) denotes the size of the smallest hub labeling (resp. smallest hierarchical hub labeling) of $G_n$. 