Quantum algorithms for Information Set Decoding

Elena Kirshanova

ENS Lyon

April 11, 2018

Information Set Decoding (ISD)

The ISD Problem

Given $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}, \mathbf{s} \in \mathbb{F}_2^{n-k}$, find $\mathbf{e} \in \mathbb{F}_2^n$ s.t. $\mathbf{He} = \mathbf{s}$

We assume we know $wt(\mathbf{e}) = w = \gamma \cdot n$, $\gamma < 1/2$

Information Set Decoding (ISD)

The ISD Problem

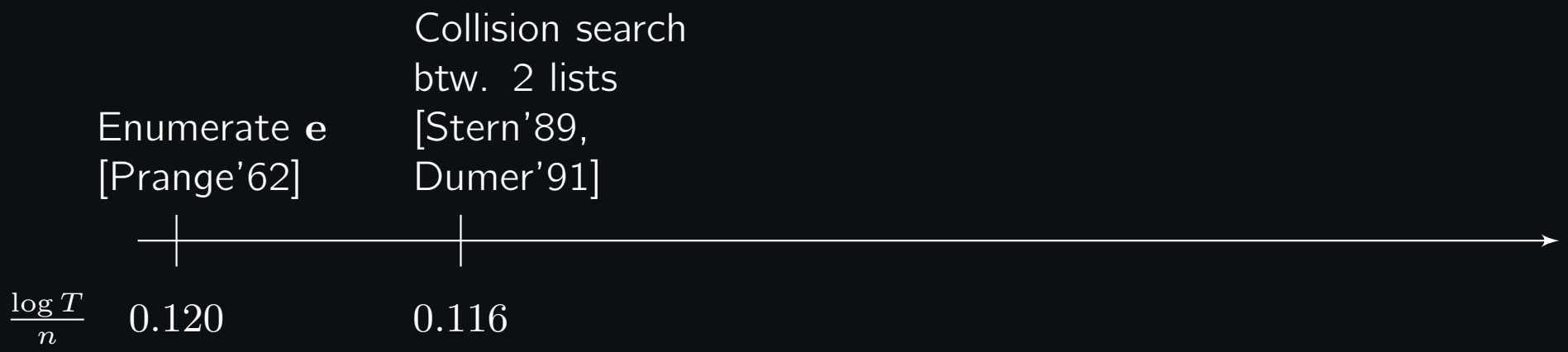
Given $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}, \mathbf{s} \in \mathbb{F}_2^{n-k}$, find $\mathbf{e} \in \mathbb{F}_2^n$ s.t. $\mathbf{He} = \mathbf{s}$

We assume we know $wt(\mathbf{e}) = w = \gamma \cdot n$, $\gamma < 1/2$

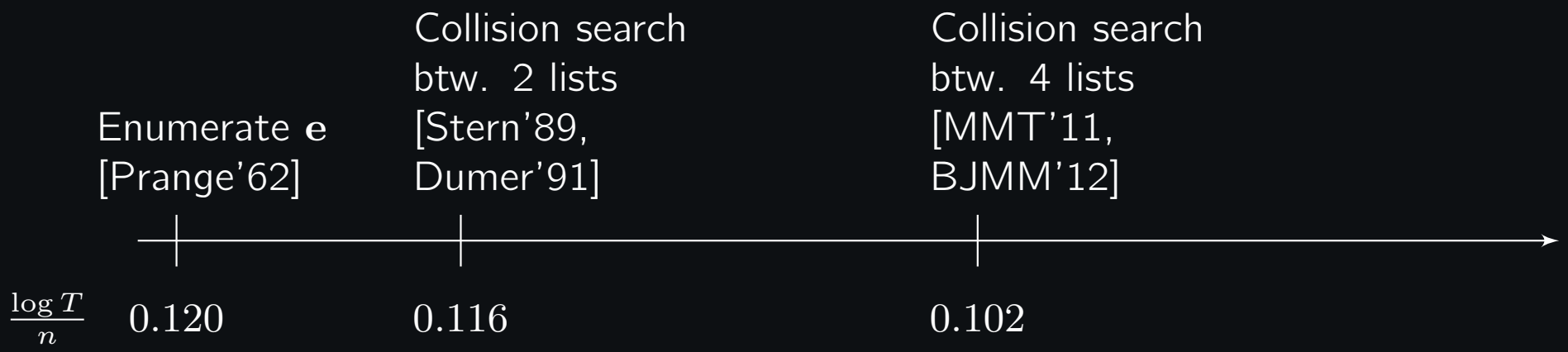
All known algorithms have complexity $2^{\mathsf{c} \cdot n + o(n)}$.
We want to improve the constant c.

Selected Algorithms for ISD (full-distance)

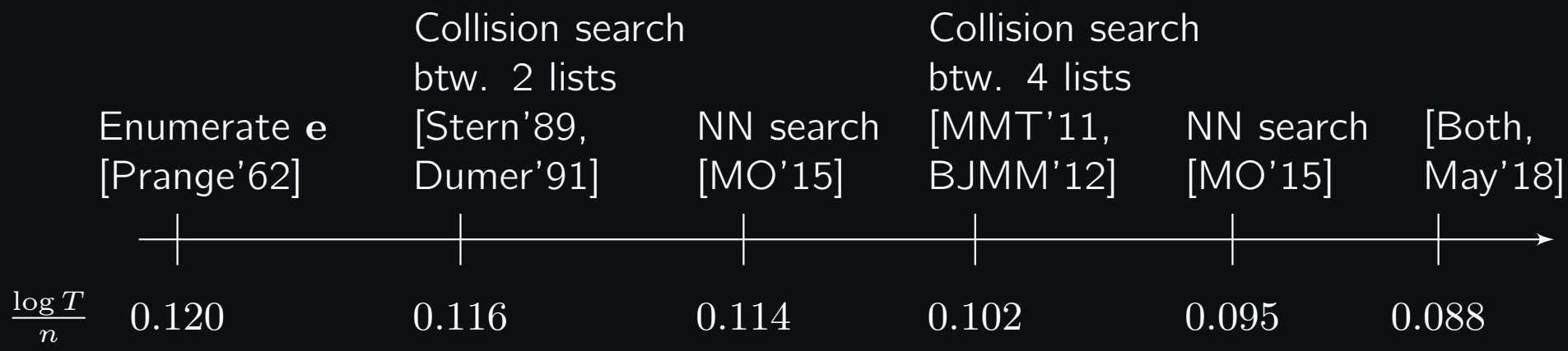
Selected Algorithms for ISD (full-distance)



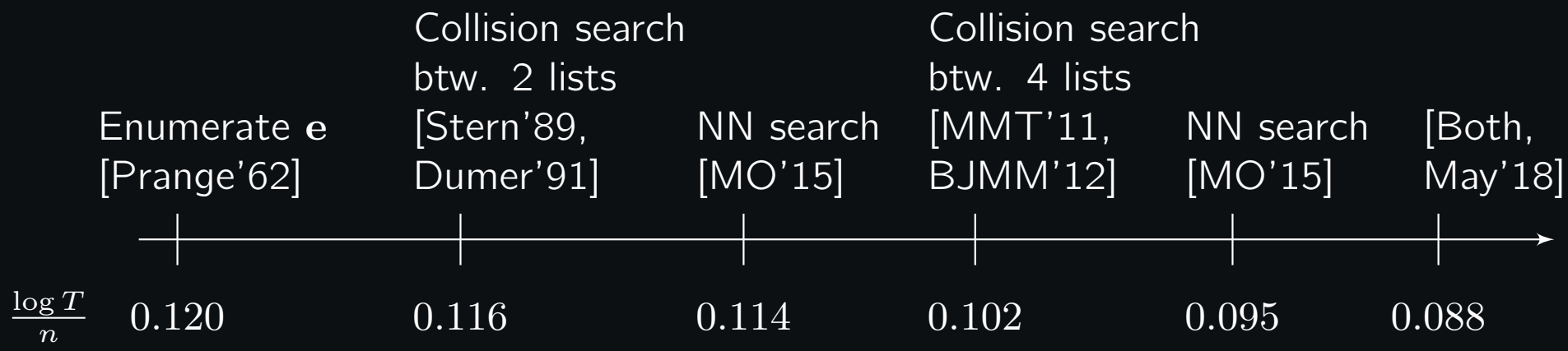
Selected Algorithms for ISD (full-distance)



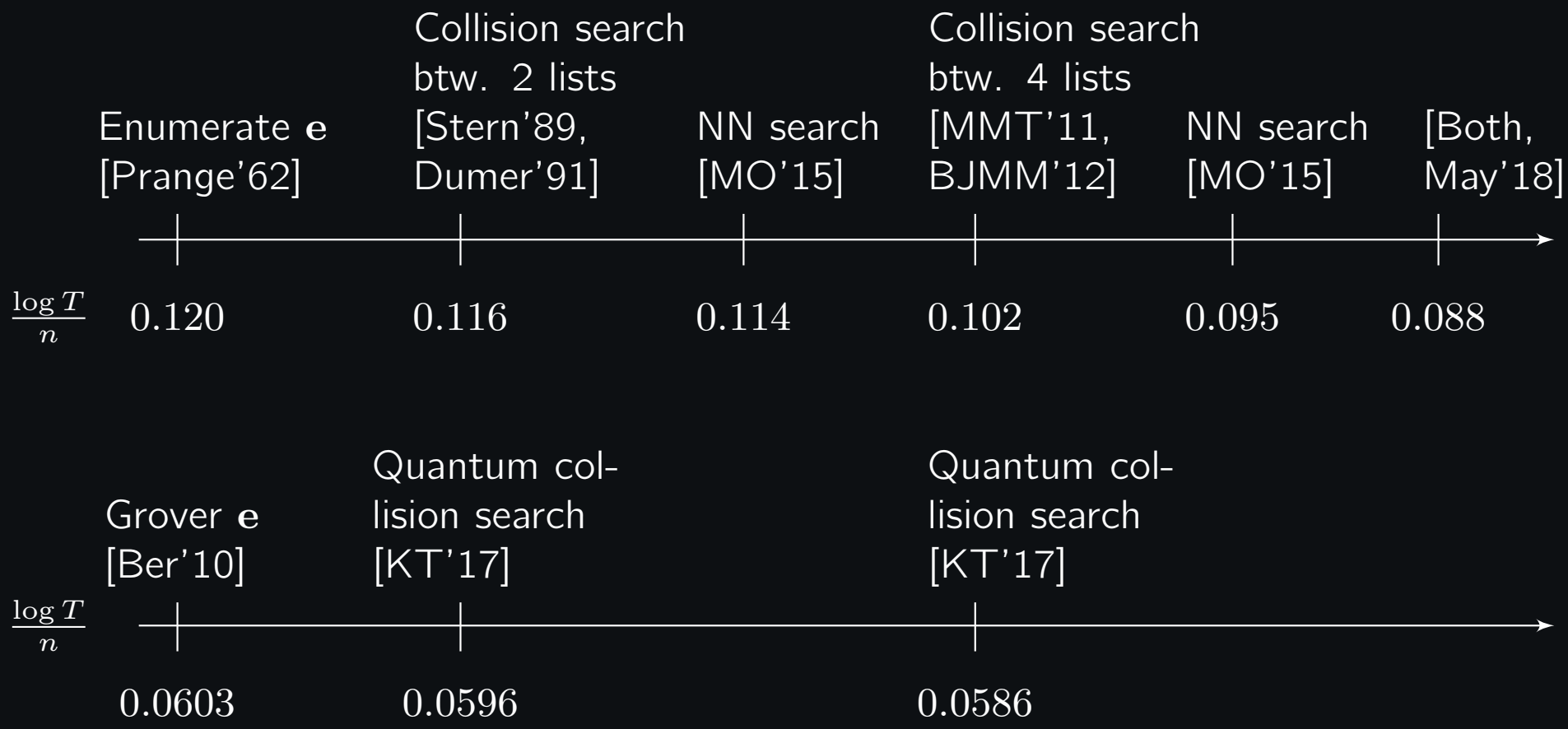
Selected Algorithms for ISD (full-distance)



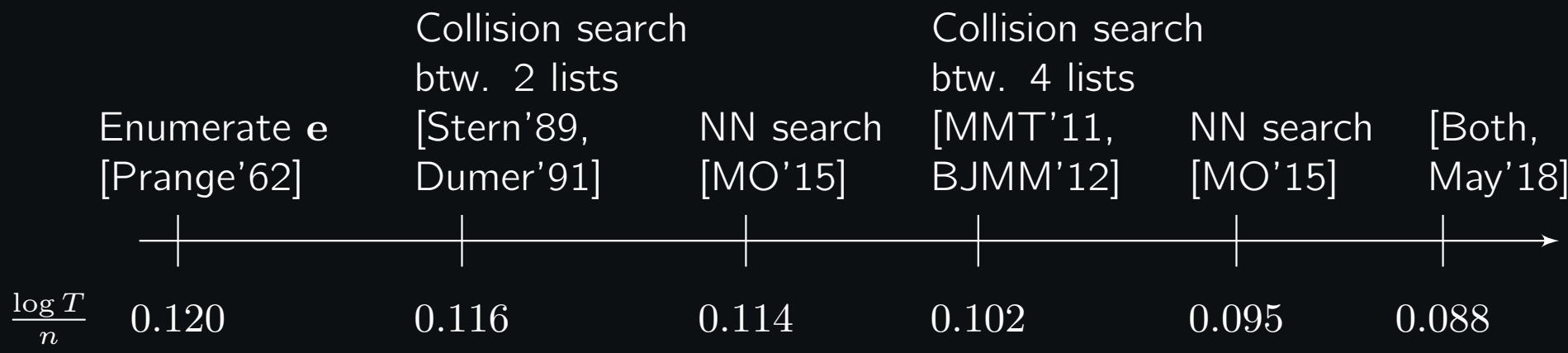
Selected Algorithms for ISD (full-distance)

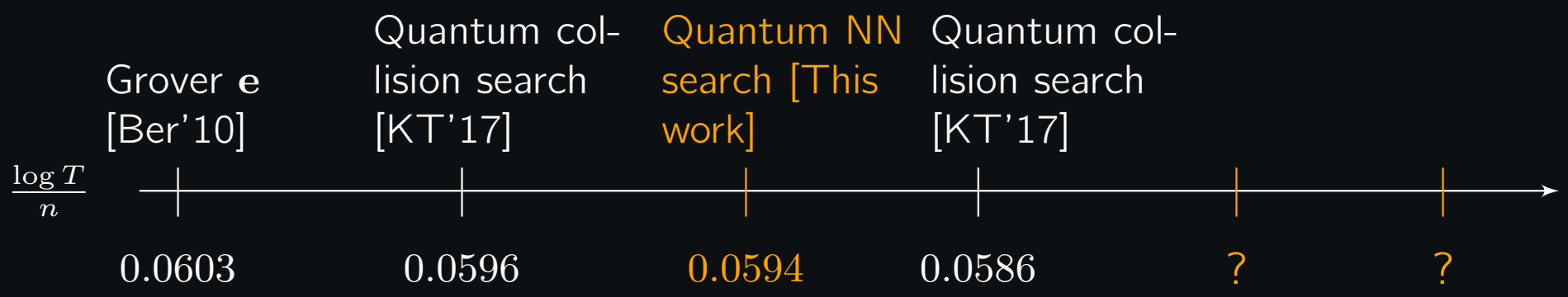


Selected Algorithms for ISD (full-distance)



Selected Algorithms for ISD (full-distance)





Main results

- An analysis of locality-sensitive filtering techniques for hamming metric
- A quantum version of May-Ozerov ISD

Algorithms for ISD: Stern'89

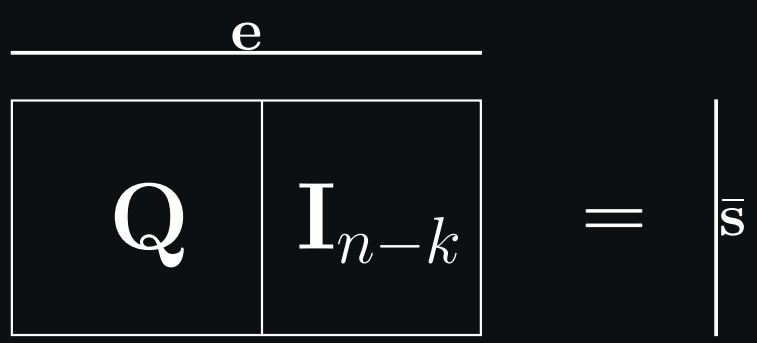
Given $\mathbf{H}, \mathbf{s}$, find $\mathbf{e}$

Algorithms for ISD: Stern'89

Given $\mathbf{H}, \mathbf{s}$, find $\mathbf{e}$

Step1: Bring $\mathbf{H}$ into the systematic form:

$$\mathbf{H} \cdot \mathbf{P} =$$



For $\bar{\mathbf{s}} = \mathbf{s} \cdot \mathbf{P}$

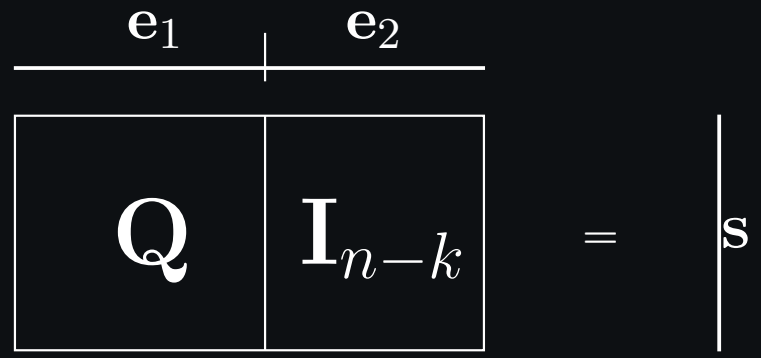
Algorithms for ISD: Stern'89

Step2: Search for collisions

$$\mathbf{Qe}_1 + \mathbf{e}_2 = \mathbf{s}$$

$$\mathbf{Qe}_1 \approx \mathbf{s}$$

$$[\mathbf{Qe}_1]_\ell = [\mathbf{s}]_\ell$$



Algorithms for ISD: Stern'89

Step2: Search for collisions

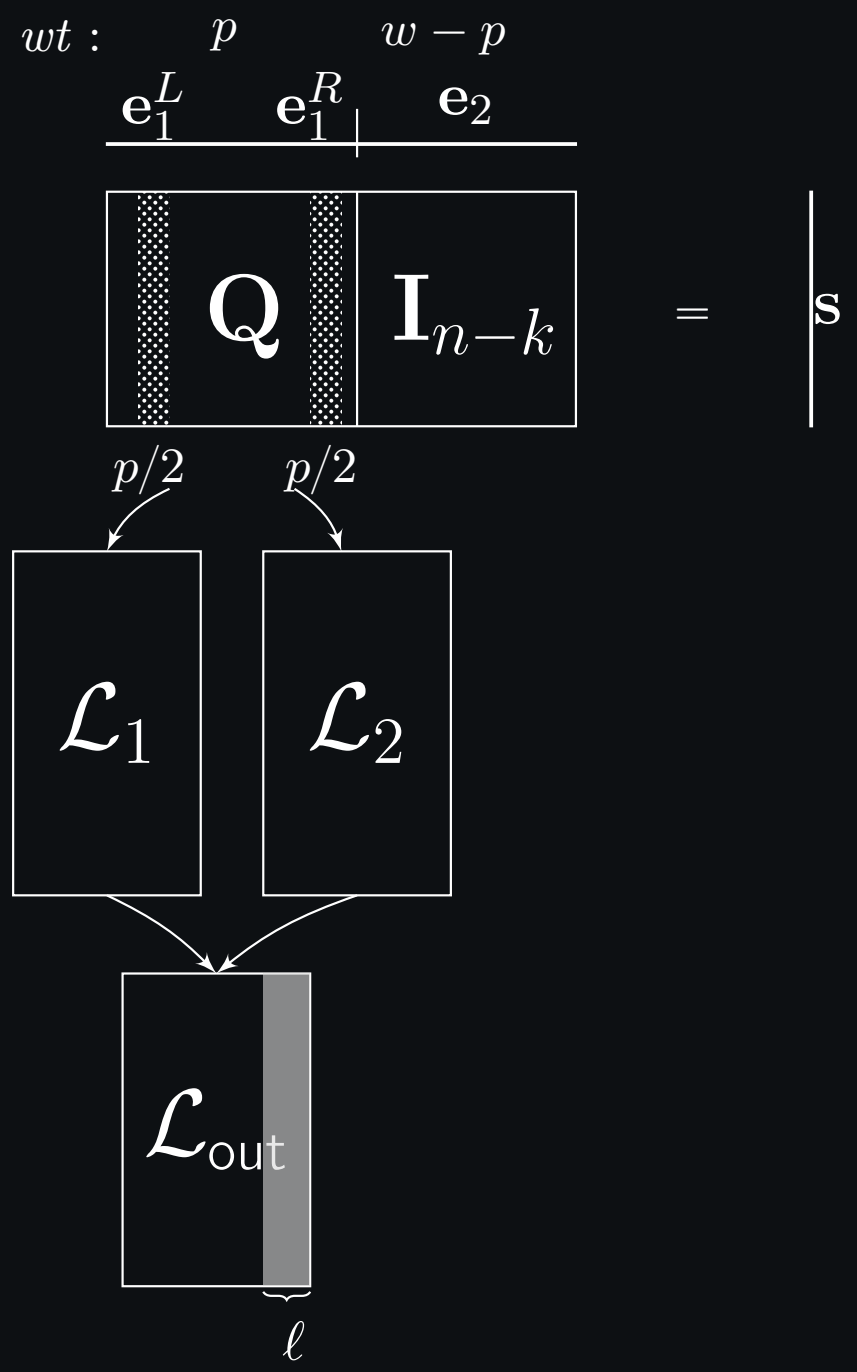
$$\mathbf{Qe}_1 + \mathbf{e}_2 = \mathbf{s}$$

$$\mathbf{Qe}_1 \approx \mathbf{s}$$

$$[\mathbf{Qe}_1]_\ell = [\mathbf{s}]_\ell$$

$$\mathcal{L}_1 = \{(\mathbf{e}_1^L, \mathbf{Qe}_1^L) : \ wt(\mathbf{e}_1^L) = \tfrac{p}{2}\}$$

$$\mathcal{L}_2 = \{(\mathbf{e}_1^R, \mathbf{Qe}_1^R + \mathbf{s}) : wt(\mathbf{e}_1^R) = \tfrac{p}{2}\}$$



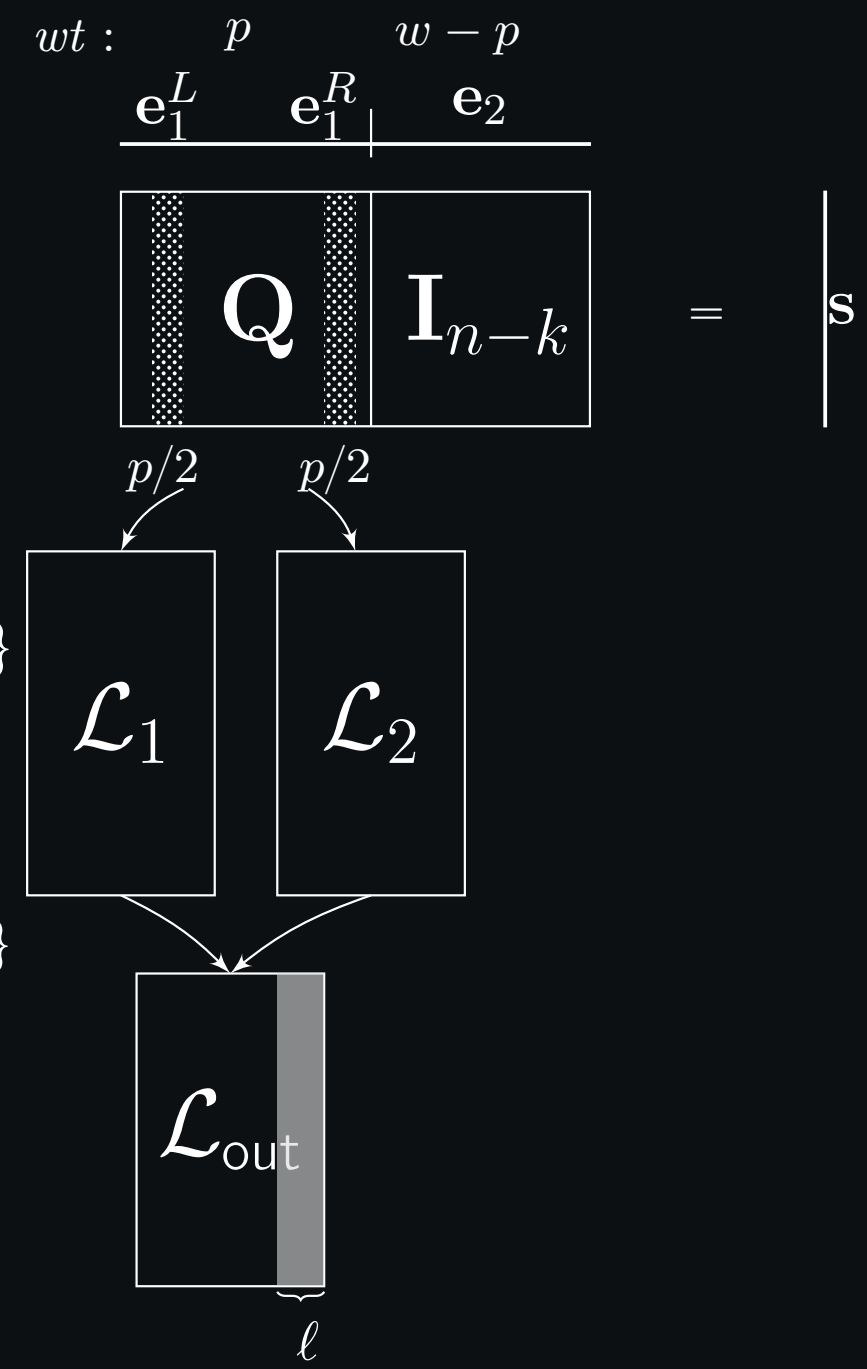
Algorithms for ISD: Stern'89

Step2: Search for collisions

$$\mathbf{Qe}_1 + \mathbf{e}_2 = \mathbf{s}$$

$$\mathbf{Qe}_1 \approx \mathbf{s}$$

$$[\mathbf{Qe}_1]_\ell = [\mathbf{s}]_\ell$$



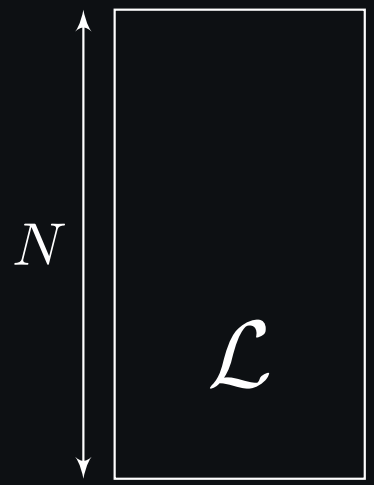
$$\mathcal{L}_1 = \{(\mathbf{e}_1^L, \mathbf{Qe}_1^L) : \ wt(\mathbf{e}_1^L) = \tfrac{p}{2}\}$$

$$\mathcal{L}_2 = \{(\mathbf{e}_1^R, \mathbf{Qe}_1^R + \mathbf{s}) : wt(\mathbf{e}_1^R) = \tfrac{p}{2}\}$$

$$\mathcal{L}_{\text{out}} = \mathcal{L}_1 \cup \mathcal{L}_2 \ : \ \mathcal{L}_1[2] = \mathcal{L}_2[2] \text{ on } \ell$$

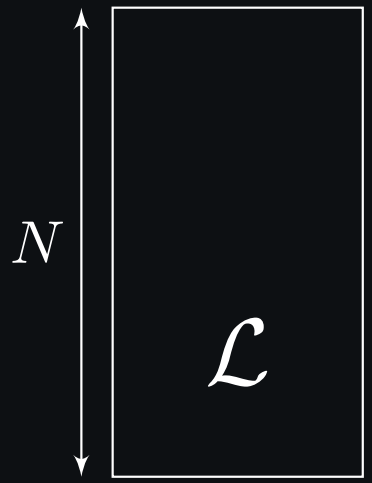
$$T = \Pr[\mathbf{e}_2 = 0 \text{ on } \ell] \cdot \max\{|\mathcal{L}_i|, |\mathcal{L}_{\text{out}}|\}$$

Quantum collision search



Goal: find $x_i = x_j, i \neq j$ - a collision in $\mathcal{L}$.

Quantum collision search

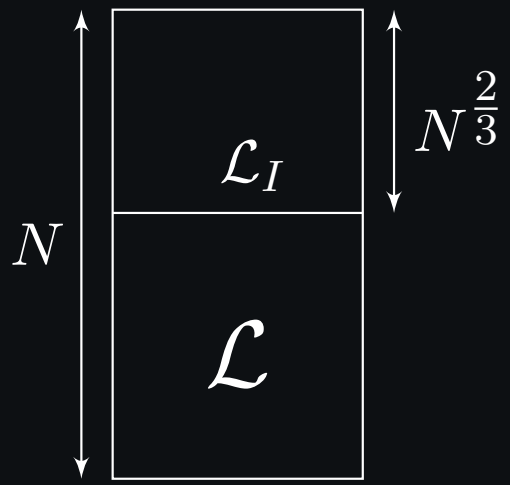


Goal: find $x_i = x_j, i \neq j$ - a collision in $\mathcal{L}$.

- create a superposition over all $N^{2/3}$-subsets of $[N]$

$$\sum_{\substack{I \subset [N] \\ |I| = N^{2/3}}} |I\rangle$$

Quantum collision search

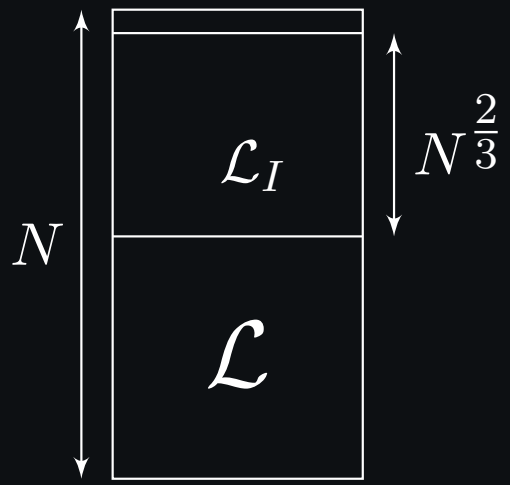


Goal: find $x_i = x_j, i \neq j$ - a collision in $\mathcal{L}$.

- create a superposition over all $N^{2/3}$-subsets of $[N]$
- 'upload' $\mathcal{L}_I$
- preprocess L_I to quickly decide whether $\mathcal{L}_I$ contains a collision

$$\sum_{\substack{I \subset [N] \\ |I| = N^{2/3}}} |I\rangle \longrightarrow \sum_{\substack{I \subset [N] \\ |I| = N^{2/3}}} |I\rangle\,|\mathcal{L}_I\rangle$$

Quantum collision search

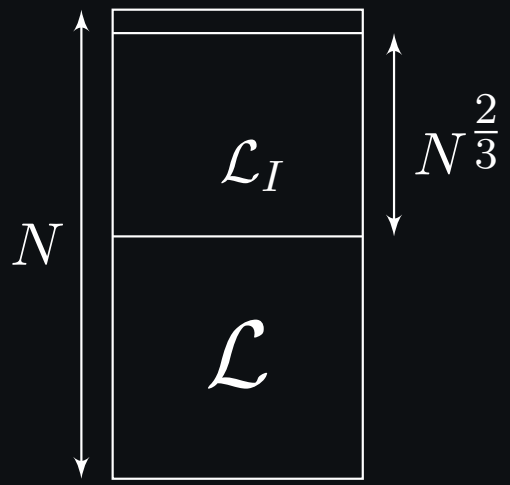


Goal: find $x_i = x_j, i \neq j$ - a collision in $\mathcal{L}$.

- create a superposition over all $N^{2/3}$-subsets of $[N]$
- 'upload' $\mathcal{L}_I$
- preprocess L_I to quickly decide whether $\mathcal{L}_I$ contains a collision
- perform quantum walk

$$\sum_{\substack{I\subset[N]\\|I|=N^{2/3}}} |I\rangle \longrightarrow \sum_{\substack{I\subset[N]\\|I|=N^{2/3}}} |I\rangle\,|\mathcal{L}_I\rangle \longrightarrow \sum_{\substack{I\subset[N]\\|I|=N^{2/3}}} |I\backslash\{i\}\cup\{j\}\rangle\,|\mathcal{L}_{I'}\rangle$$

Quantum collision search



Goal: find $x_i = x_j, i \neq j$ - a collision in $\mathcal{L}$.

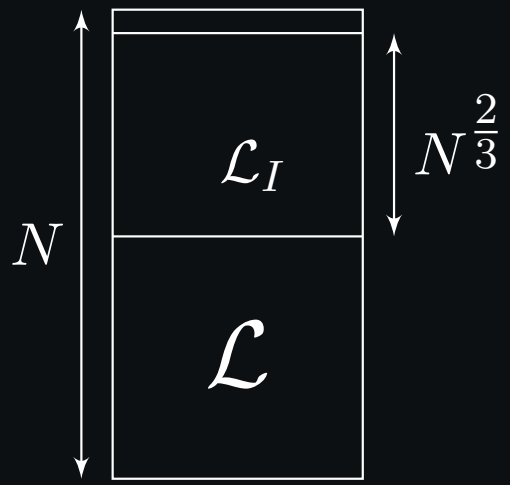
- create a superposition over all $N^{2/3}$-subsets of $[N]$
- 'upload' $\mathcal{L}_I$
- preprocess L_I to quickly decide whether $\mathcal{L}_I$ contains a collision
- perform quantum walk

$$\sum_{\substack{I\subset[N]\\|I|=N^{2/3}}} |I\rangle \longrightarrow \sum_{\substack{I\subset[N]\\|I|=N^{2/3}}} |I\rangle\,|\mathcal{L}_I\rangle \longrightarrow \sum_{\substack{I\subset[N]\\|I|=N^{2/3}}} |I\backslash\{i\}\cup\{j\}\rangle\,|\mathcal{L}_{I'}\rangle$$

$N^{1/3}$ times ↻

↺ $N^{1/3}$ times

Quantum collision search



Goal: find $x_i = x_j, i \neq j$ - a collision in $\mathcal{L}$.

- create a superposition over all $N^{2/3}$-subsets of $[N]$
- 'upload' $\mathcal{L}_I$
- preprocess L_I to quickly decide whether $\mathcal{L}_I$ contains a collision
- perform quantum walk
- measure after $\widetilde{\mathcal{O}}(N^{2/3})$ steps

$$\sum_{\substack{I\subset[N]\\|I|=N^{2/3}}} |I\rangle \longrightarrow \sum_{\substack{I\subset[N]\\|I|=N^{2/3}}} |I\rangle\,|\mathcal{L}_I\rangle \longrightarrow \sum_{\substack{I\subset[N]\\|I|=N^{2/3}}} |I\backslash\{i\}\cup\{j\}\rangle\,|\mathcal{L}_{I'}\rangle$$

$N^{1/3}$ times ↻

↺ $N^{1/3}$ times

Algorithms for ISD: Stern'89

Step2: Search for collisions

$$\mathbf{Q}\mathbf{e}_1 + \mathbf{e}_2 = \mathbf{s}$$

$$\mathbf{Q}\mathbf{e}_1 \approx \mathbf{s}$$

$$[\mathbf{Q}\mathbf{e}_1]_\ell = [\mathbf{s}]_\ell$$

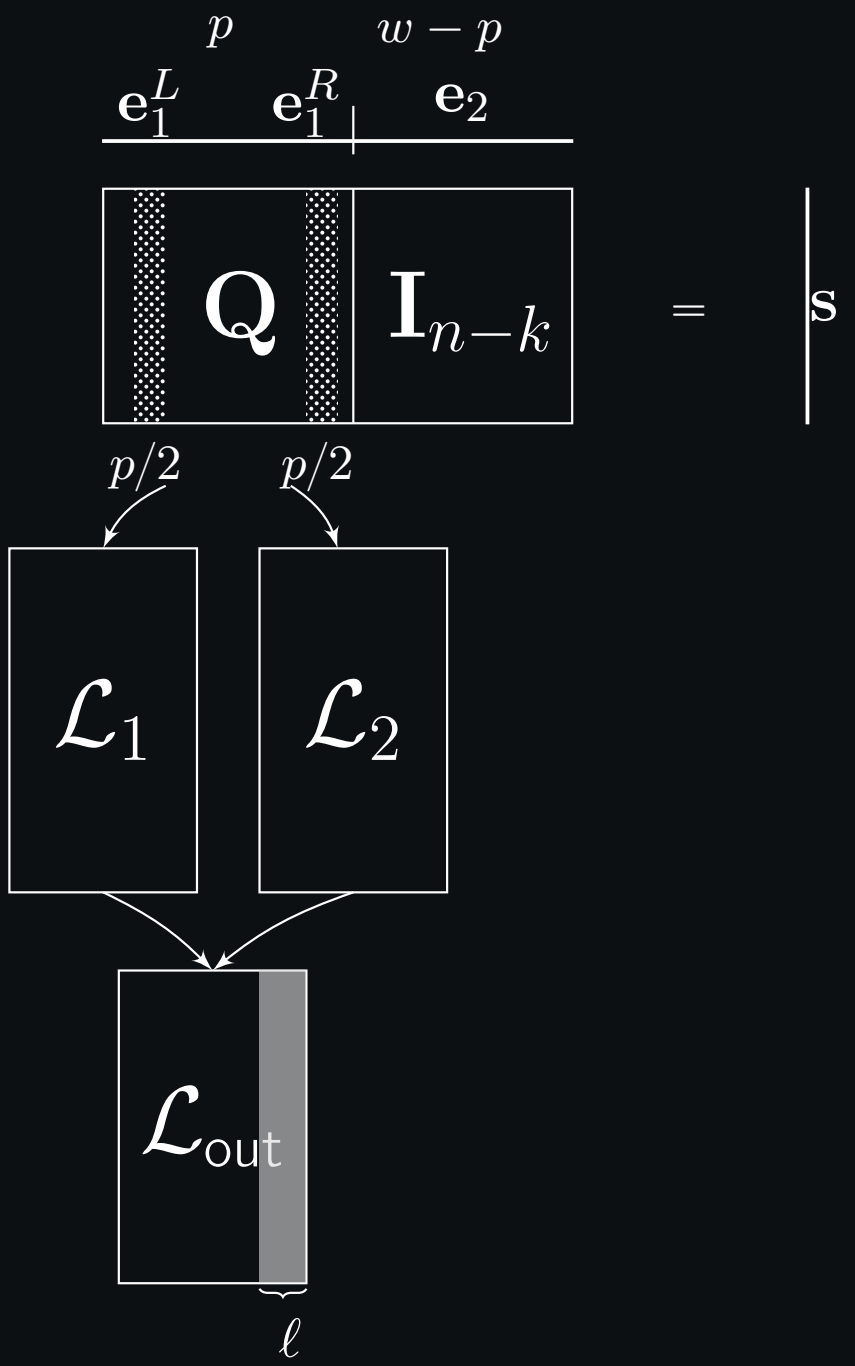
$$\mathcal{L}_1 = \{(\mathbf{e}_1^L, \mathbf{Q}\mathbf{e}_1^L) : \ wt(\mathbf{e}_1^L) = \tfrac{p}{2}\}$$

$$\mathcal{L}_2 = \{(\mathbf{e}_1^R, \mathbf{Q}\mathbf{e}_1^R + \mathbf{s}) : wt(\mathbf{e}_1^R) = \tfrac{p}{2}\}$$

$$\mathcal{L}_{\text{out}} = \mathcal{L}_1 \cup \mathcal{L}_2 \ : \ \mathcal{L}_1[2] = \mathcal{L}_2[2] \text{ on } \ell$$

$$T = \Pr[\mathbf{e}_2 = 0 \text{ on } \ell] \cdot \max\{|\mathcal{L}_i|, |\mathcal{L}_{\text{out}}|\}$$

$$T^{\mathsf{Q}} = \Pr[\mathbf{e}_2 = 0 \text{ on } \ell]^{1/2} \cdot |\mathcal{L}_i|^{2/3}$$



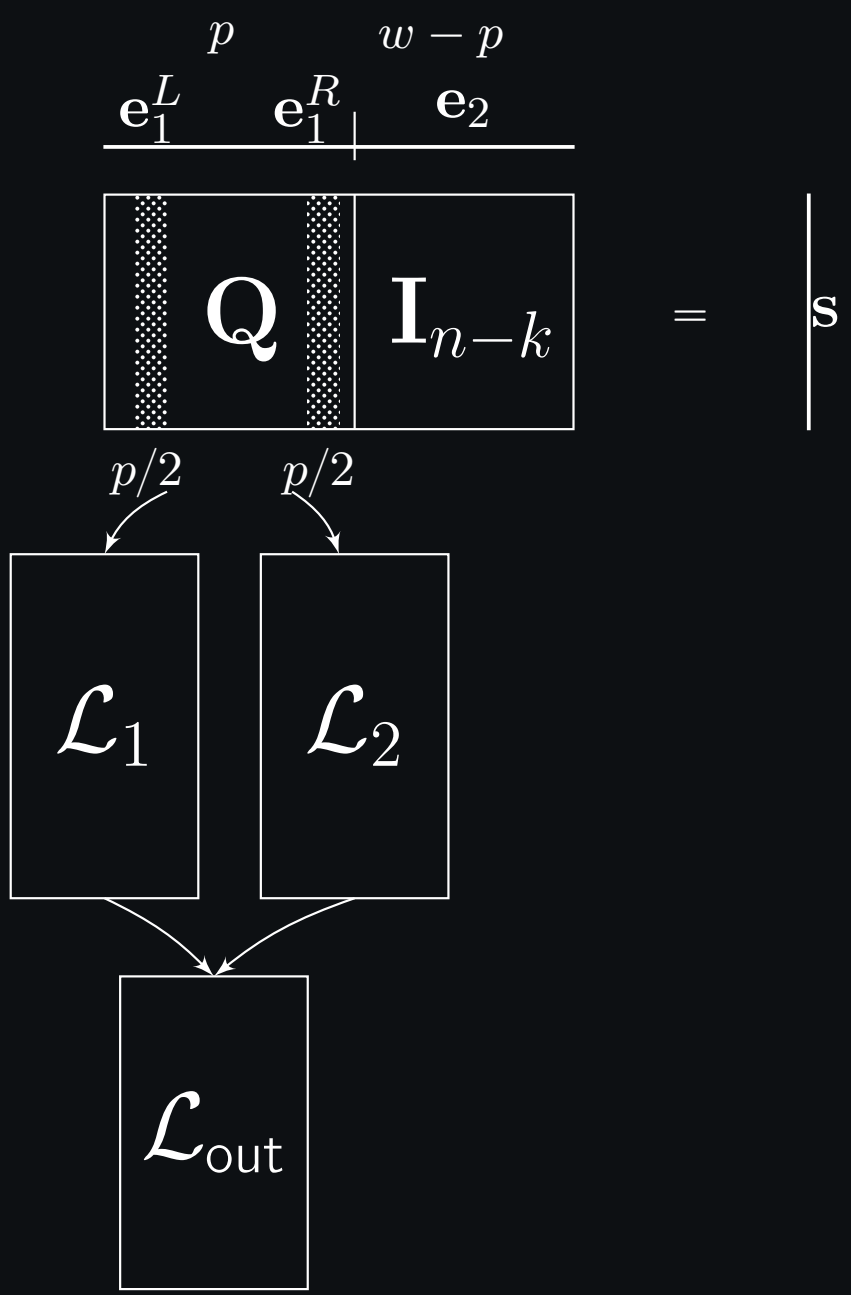
Algorithms for ISD: MO'15

Step2: Search for approximate collisions

$$\mathbf{Q}\mathbf{e}_1 + \mathbf{e}_2 = \mathbf{s}$$
$$\mathbf{Q}\mathbf{e}_1 \approx \mathbf{s}$$

$$\mathcal{L}_1 = \{(\mathbf{e}_1^L, \mathbf{Q}\mathbf{e}_1^L) : \ wt(\mathbf{e}_1^L) = \tfrac{p}{2}\}$$
$$\mathcal{L}_2 = \{(\mathbf{e}_1^R, \mathbf{Q}\mathbf{e}_1^R + \mathbf{s}) \ wt(\mathbf{e}_1^R) = \tfrac{p}{2}\}$$

$$\mathcal{L}_{\text{out}} = \mathcal{L}_1 \cup \mathcal{L}_2 \text{ s.t. } \mathcal{L}_1[2] \approx \mathcal{L}_2[2]$$
$$T = T_{\text{NN}}(|\mathcal{L}_i|, w)$$



Locality sensitive filtering over $\mathbb{F}_2$

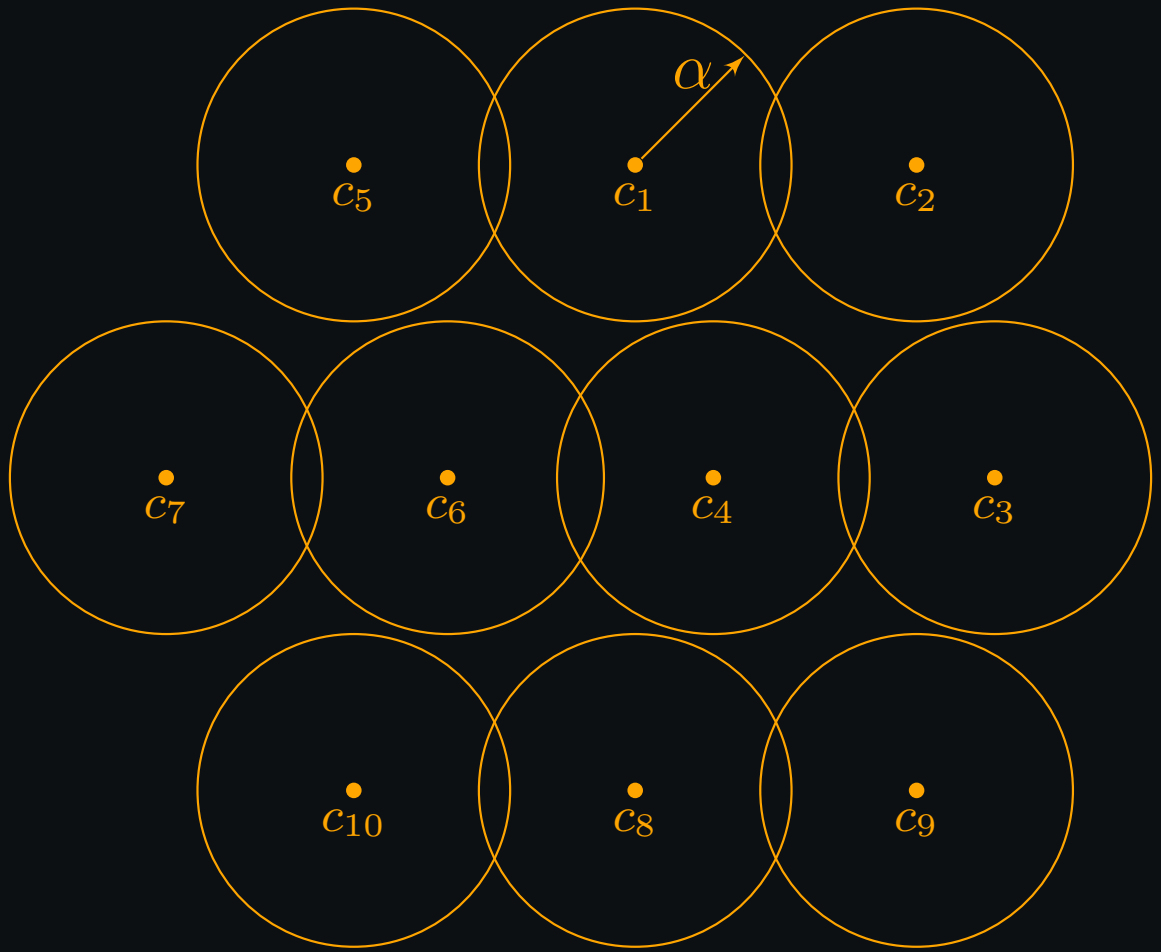
γ-Near Neighbour

Given $\mathcal{L} \subset \mathbb{F}_2^n$, preprocess $\mathcal{L}$ s.t. upon receiving a query vector $\mathbf{q} \in \mathbb{F}_2^n$, we can efficiently find all $\mathbf{v} \in \mathcal{L}$ s.t. $\mathsf{dist}(\mathbf{v}, \mathbf{q}) \leqslant \gamma \cdot n$ for $\gamma < 1/2$.

Locality sensitive filtering over $\mathbb{F}_2$

γ-Near Neighbour

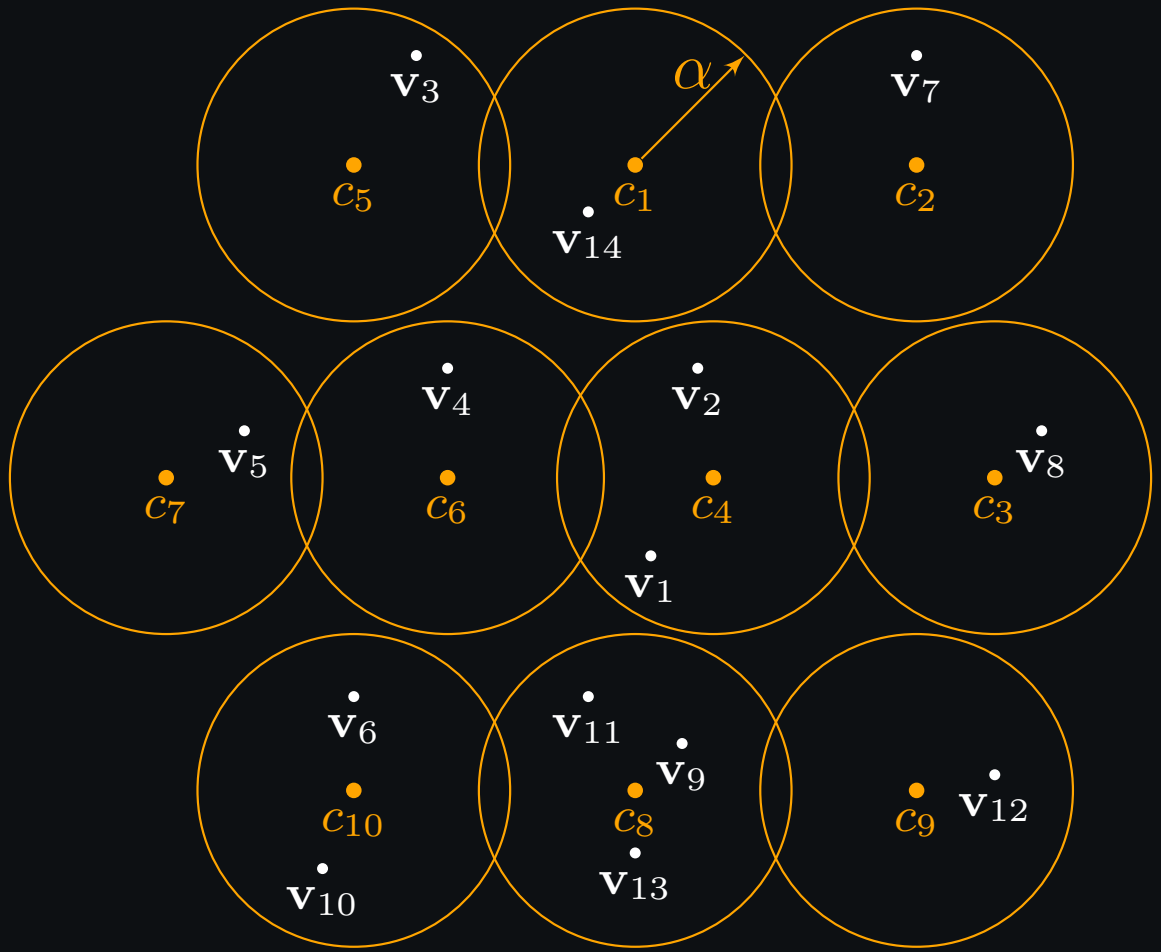
Given $\mathcal{L} \subset \mathbb{F}_2^n$, preprocess $\mathcal{L}$ s.t. upon receiving a query vector $\mathbf{q} \in \mathbb{F}_2^n$, we can efficiently find all $\mathbf{v} \in \mathcal{L}$ s.t. $\mathsf{dist}(\mathbf{v}, \mathbf{q}) \leqslant \gamma \cdot n$ for $\gamma < 1/2$.



Locality sensitive filtering over $\mathbb{F}_2$

γ-Near Neighbour

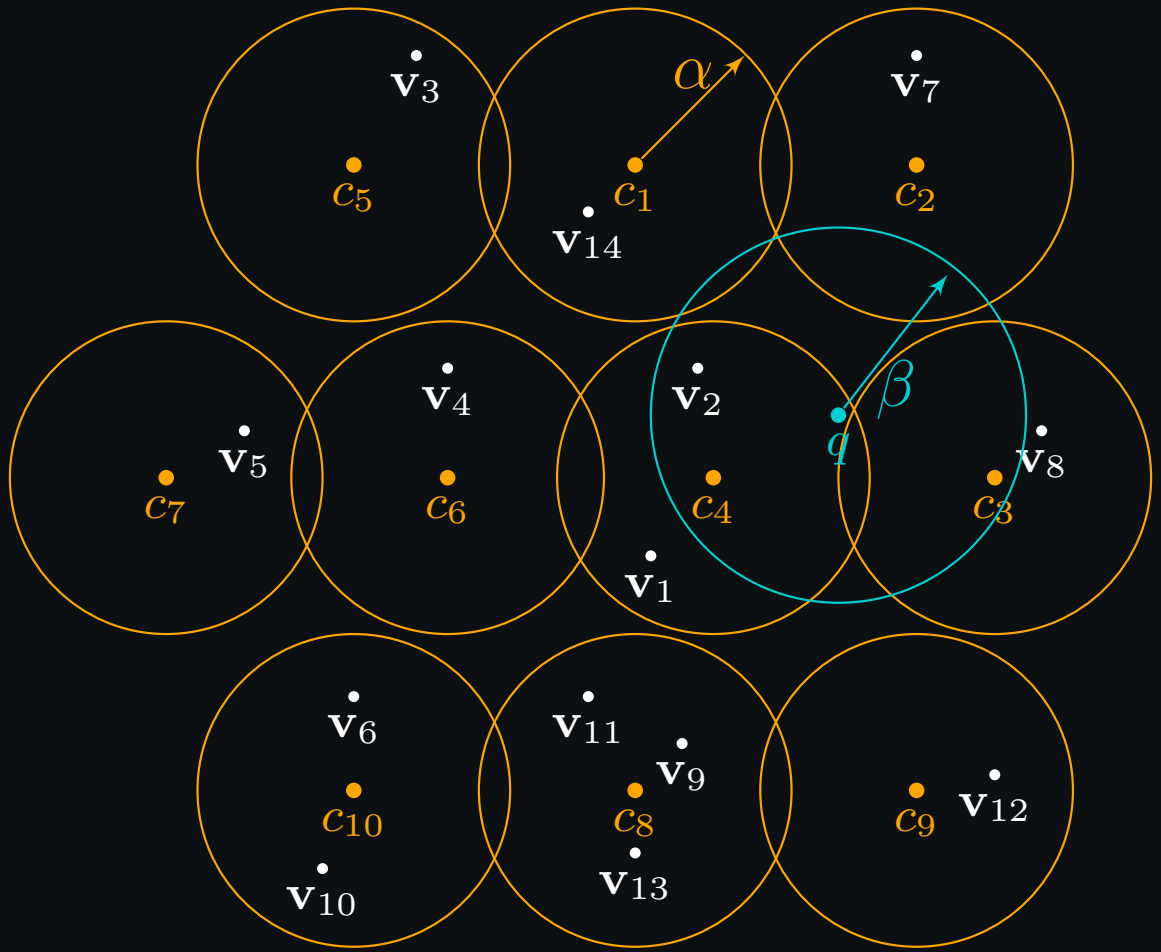
Given $\mathcal{L} \subset \mathbb{F}_2^n$, preprocess $\mathcal{L}$ s.t. upon receiving a query vector $\mathbf{q} \in \mathbb{F}_2^n$, we can efficiently find all $\mathbf{v} \in \mathcal{L}$ s.t. $\mathsf{dist}(\mathbf{v}, \mathbf{q}) \leqslant \gamma \cdot n$ for $\gamma < 1/2$.



Locality sensitive filtering over $\mathbb{F}_2$

γ-Near Neighbour

Given $\mathcal{L} \subset \mathbb{F}_2^n$, preprocess $\mathcal{L}$ s.t. upon receiving a query vector $\mathbf{q} \in \mathbb{F}_2^n$, we can efficiently find all $\mathbf{v} \in \mathcal{L}$ s.t. $\mathsf{dist}(\mathbf{v}, \mathbf{q}) \leqslant \gamma \cdot n$ for $\gamma < 1/2$.

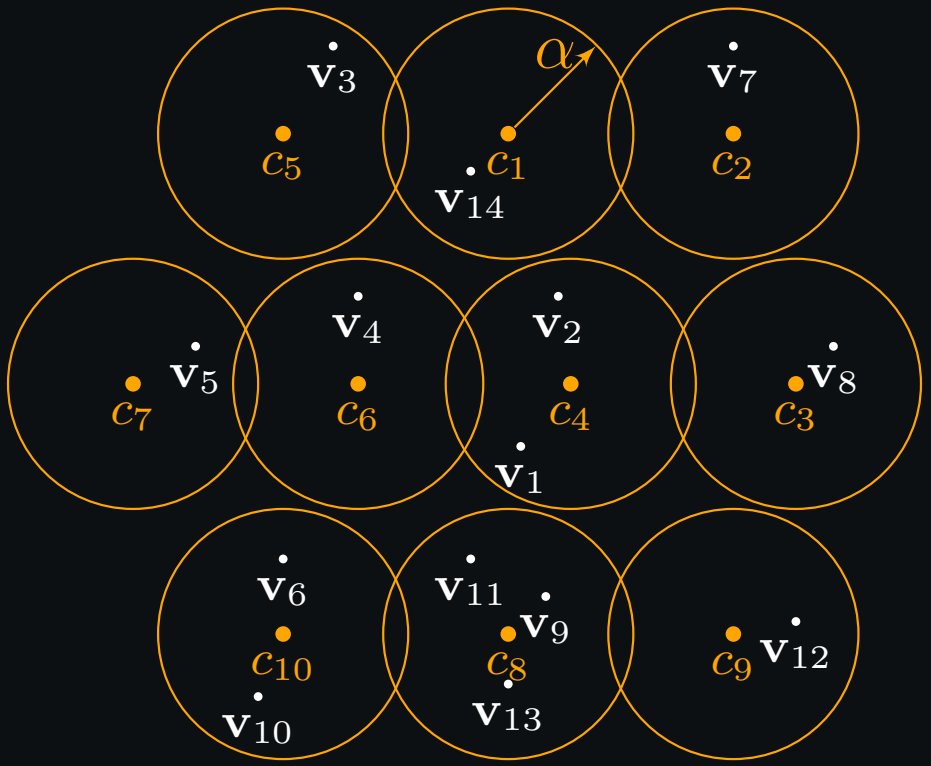


Classical LSF

Algorithms

$\mathcal{D}$ - the LSF data structure: $\mathcal{D} = \cup_{c \in \mathcal{C}} \text{Bucket}_c$

Classical LSF

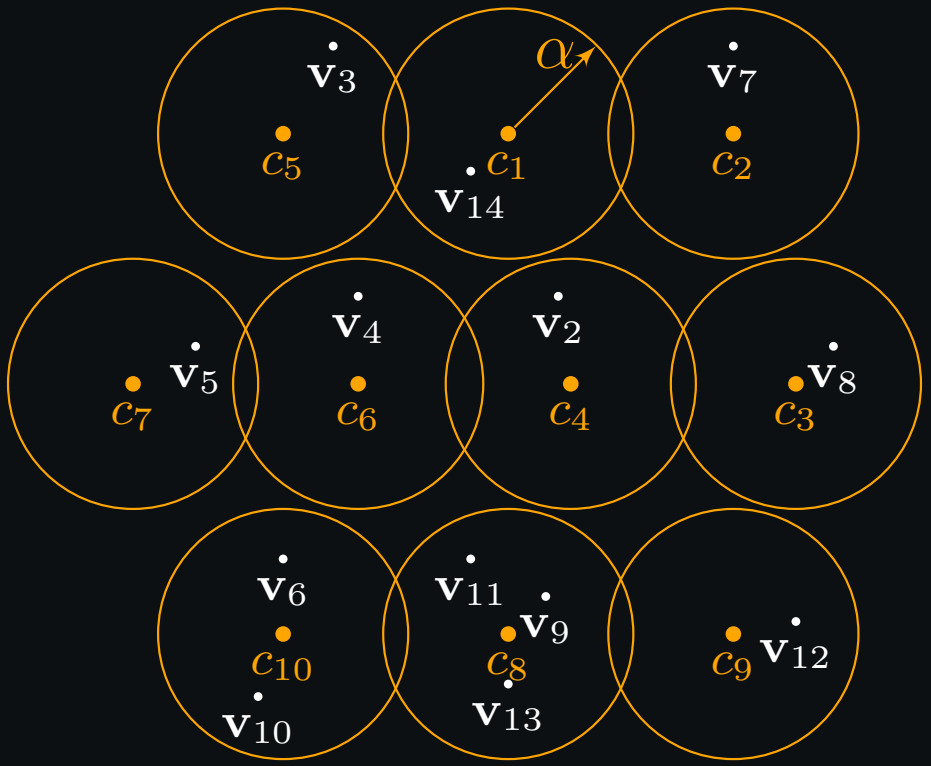


Algorithms

$\mathcal{D}$ - the LSF data structure: $\mathcal{D} = \cup_{c \in \mathcal{C}} \text{Bucket}_c$

$\mathcal{D}.\textbf{Insert}^{\alpha}(\mathbf{v})$: Add $\mathbf{v}$ to all the relevant buckets of $\mathcal{D}$

Classical LSF



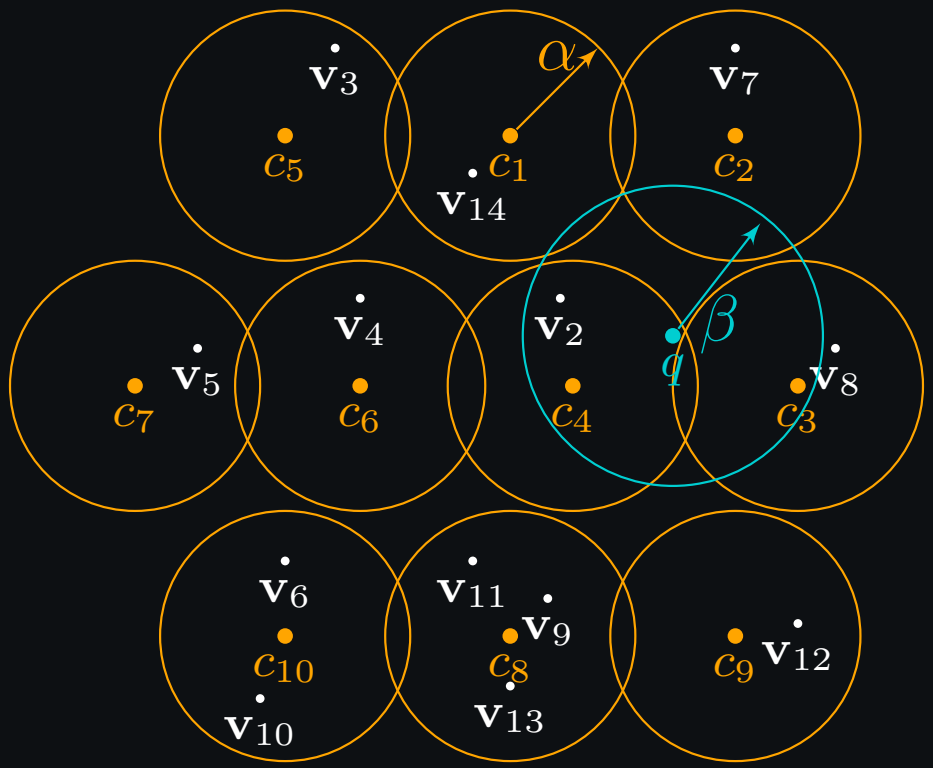
Algorithms

$\mathcal{D}$ - the LSF data structure: $\mathcal{D} = \cup_{c \in \mathcal{C}} \text{Bucket}_c$

$\mathcal{D}.\textbf{Insert}^{\alpha}(\mathbf{v})$: Add $\mathbf{v}$ to all the relevant buckets of $\mathcal{D}$

$\mathcal{D}.\textbf{Remove}^{\alpha}(\mathbf{v})$: Remove $\mathbf{v}$ from all buckets

Classical LSF



Algorithms

$\mathcal{D}$ - the LSF data structure: $\mathcal{D} = \cup_{c \in \mathcal{C}} \text{Bucket}_c$

$\mathcal{D}.\textbf{Insert}^{\alpha}(\mathbf{v})$: Add $\mathbf{v}$ to all the relevant buckets of $\mathcal{D}$

$\mathcal{D}.\textbf{Remove}^{\alpha}(\mathbf{v})$: Remove $\mathbf{v}$ from all buckets

$\mathcal{D}.\textbf{Query}^{\beta}(\mathbf{q})$: Find all $\mathbf{v} \in \mathcal{D}$ with $\textbf{dist}(\mathbf{v}, \mathbf{q}) \leqslant \beta$

We have runtimes of these algorithms.

LSF for Quantum walk

Quantum walk

use some data structure $\mathcal{D}$
to check for
$=$

Quantum walk with NN

use `LSF`-data structure
$\mathcal{D} = \cup_c \text{Bucket}_c$ to check for
$\approx$

LSF for Quantum walk

Quantum walk

use some data structure $\mathcal{D}$
to check for
$=$

Quantum walk with NN

use `LSF`-data structure
$\mathcal{D} = \cup_c \text{Bucket}_c$ to check for
$\approx$

Setup:

$$\sum_{\substack{I \subset [N] \\ |I| = N^{2/3}}} |I\rangle |\mathcal{L}_I\rangle$$

$$\sum_{\substack{I \subset [N] \\ |I| = N^{2/3}}} |I\rangle |\mathcal{D}.\texttt{Update}(L_I)\rangle$$

LSF for Quantum walk

Quantum walk

use some data structure $\mathcal{D}$
to check for
$=$

Quantum walk with NN

use `LSF`-data structure
$\mathcal{D} = \cup_c \text{Bucket}_c$ to check for
$\approx$

Setup:

$$\sum_{\substack{I \subset [N] \\ |I| = N^{2/3}}} |I\rangle |\mathcal{L}_I\rangle$$

$$\sum_{\substack{I \subset [N] \\ |I| = N^{2/3}}} |I\rangle |\mathcal{D}.\texttt{Update}(L_I)\rangle$$

A step of the walk:

$$\sum |I \backslash \{x\}\rangle$$
$$\sum |I \cup \{y\}\rangle$$

$$\sum |I\rangle |\mathcal{D}.\texttt{Remove}(x)\rangle$$
$$\sum |I\rangle |\mathcal{D}.\texttt{Update}(y)\rangle$$

LSF for Quantum walk

Quantum walk		Quantum walk with NN
use some data structure $\mathcal{D}$ to check for $=$		use `LSF`-data structure $\mathcal{D} = \cup_c \text{Bucket}_c$ to check for $\approx$
$\sum_{\substack{I\subset[N]\\|I|=N^{2/3}}} \lvert I\rangle \lvert\mathcal{L}_I\rangle$	Setup:	$\sum_{\substack{I\subset[N]\\|I|=N^{2/3}}} \lvert I\rangle \lvert\mathcal{D}.\texttt{Update}(L_I)\rangle$
$\sum \lvert I\backslash\{x\}\rangle$ $\sum \lvert I\cup\{y\}\rangle$	A step of the walk:	$\sum \lvert I\rangle \lvert\mathcal{D}.\texttt{Remove}(x)\rangle$ $\sum \lvert I\rangle \lvert\mathcal{D}.\texttt{Update}(y)\rangle$
$\sum \lvert I\rangle \left\lvert \mathbf{v}_i \stackrel{?}{=} y \right\rangle$	Check for (approximate) equality:	$\sum \lvert I\rangle \lvert\mathcal{D}.\texttt{Query}(y)\rangle$

LSF for Quantum walk

Quantum walk

use some data structure $\mathcal{D}$ to check for $=$

Setup:

$$\sum_{\substack{I\subset[N]\\|I|=N^{2/3}}} |I\rangle|\mathcal{L}_I\rangle$$

A step of the walk:

$$\sum |I\backslash\{x\}\rangle$$
$$\sum |I\cup\{y\}\rangle$$

Check for (approximate) equality:

$$\sum |I\rangle\left|\mathbf{v}_i \stackrel{?}{=} y\right\rangle$$

$$\log(T) = 0.0597\cdot n$$

Quantum walk with NN

use `LSF`-data structure $\mathcal{D} = \cup_c \text{Bucket}_c$ to check for $\approx$

Setup:

$$\sum_{\substack{I\subset[N]\\|I|=N^{2/3}}} |I\rangle|\mathcal{D}.\texttt{Update}(L_I)\rangle$$

A step of the walk:

$$\sum |I\rangle|\mathcal{D}.\texttt{Remove}(x)\rangle$$
$$\sum |I\rangle|\mathcal{D}.\texttt{Update}(y)\rangle$$

Check for (approximate) equality:

$$\sum |I\rangle|\mathcal{D}.\texttt{Query}(y)\rangle$$

$$\log(T) = 0.0594\cdot n$$

LSF for Quantum walk

Quantum walk		Quantum walk with NN
use some data structure $\mathcal{D}$ to check for $=$	Setup:	use `LSF`-data structure $\mathcal{D} = \cup_c \text{Bucket}_c$ to check for $\approx$
$\sum_{\substack{I\subset[N]\\|I|=N^{2/3}}} \lvert I\rangle\lvert\mathcal{L}_I\rangle$		$\sum_{\substack{I\subset[N]\\|I|=N^{2/3}}} \lvert I\rangle\lvert\mathcal{D}.\texttt{Update}(L_I)\rangle$
$\sum \lvert I\backslash\{x\}\rangle$ $\sum \lvert I\cup\{y\}\rangle$	A step of the walk:	$\sum \lvert I\rangle\lvert\mathcal{D}.\texttt{Remove}(x)\rangle$ $\sum \lvert I\rangle\lvert\mathcal{D}.\texttt{Update}(y)\rangle$
$\sum \lvert I\rangle\left\lvert \mathbf{v}_i \stackrel{?}{=} y\right\rangle$	Check for (approximate) equality:	$\sum \lvert I\rangle\lvert\mathcal{D}.\texttt{Query}(y)\rangle$
$\log(T) = 0.0597 \cdot n$		$\log(T) = 0.0594 \cdot n$

Thank you!