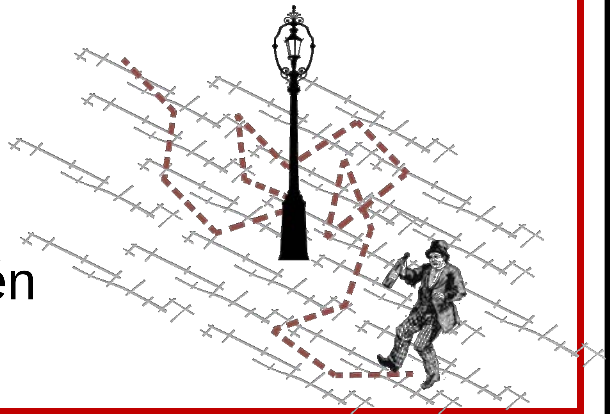
Quantum Walk Search

Stacey Jeffery

Based in part on joint work with:

Andris Ambainis
András Gilyén
Martins Kokainis

Simon Apers
András Gilyén



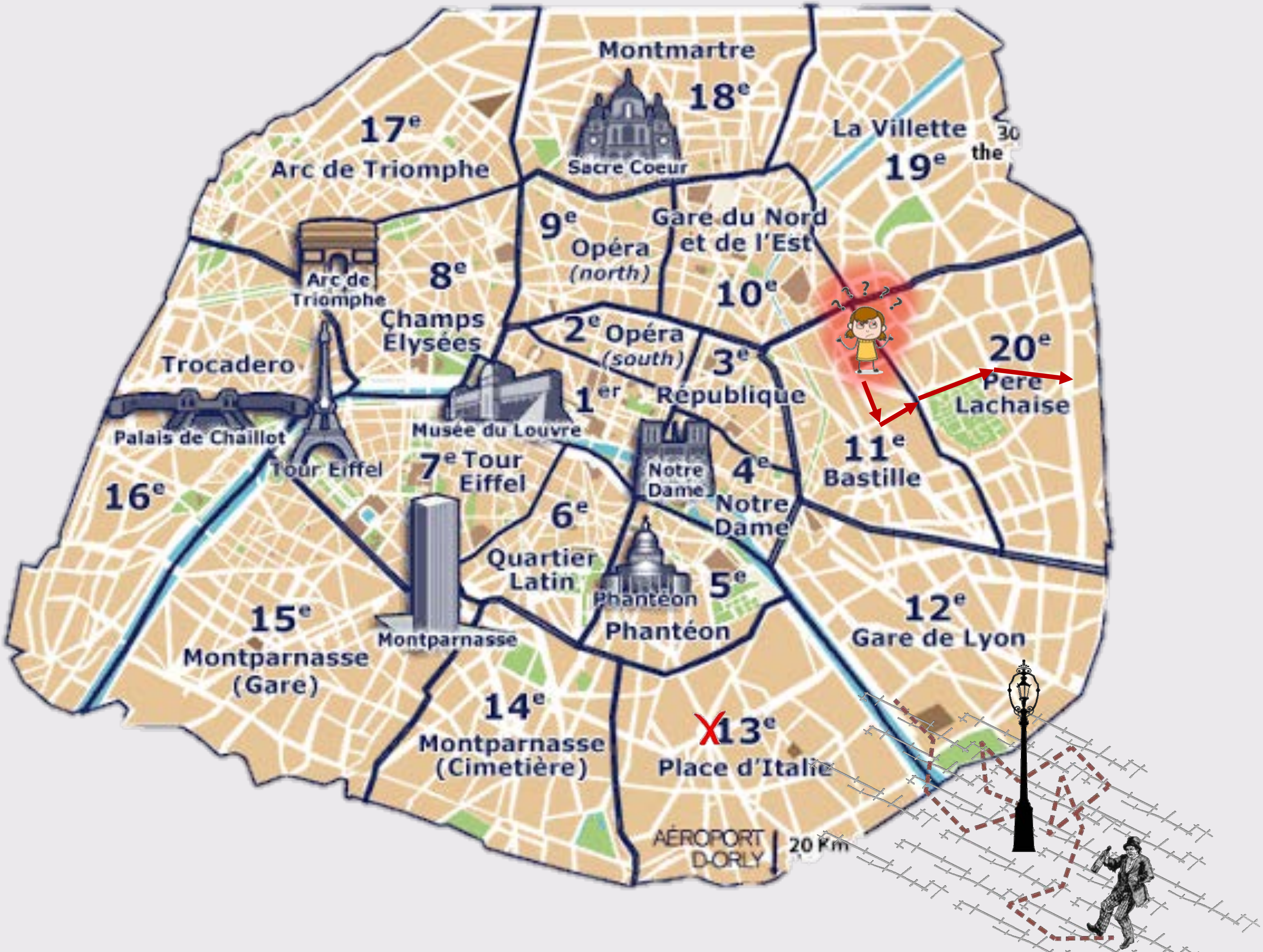


Photo credit: thescilife.org

Random Walks

in $\{0,1\}^n$ a bit 0 or 1

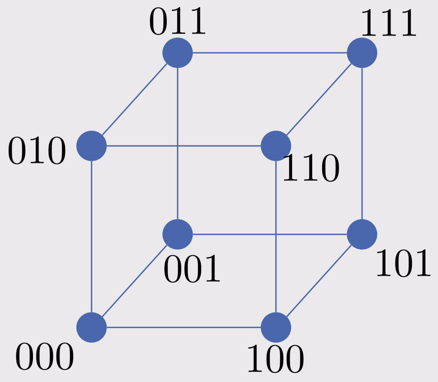
A Formula: $\phi(x) = C_1(x_1, x_3, x_7)C_2(x_2, x_5, x_6)C_3(x_1, x_2, x_4)$

Find x such that $\phi(x) = 1$

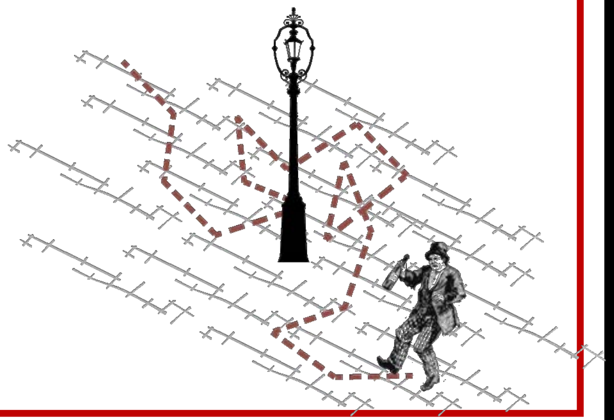
$x = 0010110 \qquad \phi(x) = 0$

$x = 0011110 \qquad \phi(x) = 0$

$x = 0011100 \qquad \phi(x) = 0$



Quantum Walk Frameworks



Framework 0: Amplitude Amplification

Algorithmic Template

```
Repeat 1/ε times:
    Sample u
    Check u
    If u marked, output
```

Complexity: $\frac{1}{\varepsilon}(\mathtt{S}+\mathtt{C})$

Quantum algorithm
with complexity
$\frac{1}{\sqrt{\varepsilon}}(\mathtt{S}+\mathtt{C})$

Subroutines

S **Sample:** Sample u, according to π

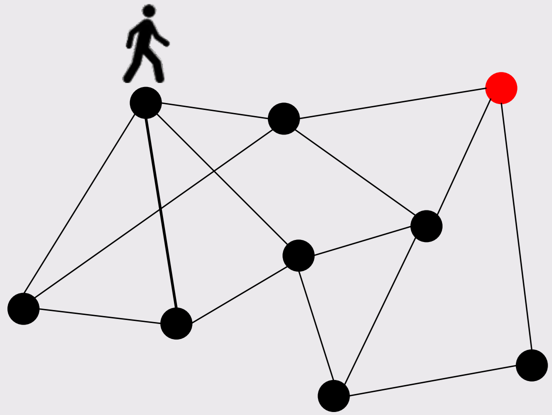
C **Check:** If at u, check if u is marked

$\varepsilon =$ probability $u \sim \pi$ marked

Framework 1: Hitting Time Framework

Algorithmic Template


`Sample` $u \sim \pi$



random walk P

Subroutines

Sample: Sample u, according to π

$P[u,v]$ = probability walker at u moves to v

π = stationary distribution of P

[Szegedy 2004] [Ambainis, Gilyén, J, Kokainis 2019]

Framework 1: Hitting Time Framework

Algorithmic Template

```
Sample u ~ π
Repeat H times:
    Update to u
    Check u
    If u marked, output
```

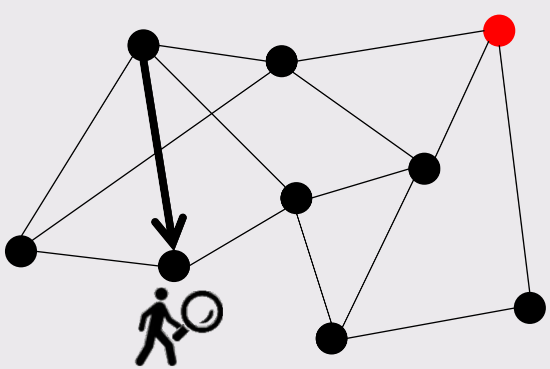
Complexity: $\mathtt{S} + H(\mathtt{U} + \mathtt{C})$

Subroutines

S — **Sample:** Sample u, according to π

U — **Update:** If at u, sample v from $P[u, \cdot]$

C — **Check:** If at u, check if u is marked



$P[u, v]$ = probability walker at u moves to v

π = stationary distribution of P

H = hitting time

[Szegedy 2004] [Ambainis, Gilyén, J, Kokainis 2019]

Framework 1: Hitting Time Framework

Algorithmic Template

`Sample` $u \sim \pi$

`Repeat` H `times:`

  `Update to` u

  `Check` u

  `If` u `marked, output`

Complexity: $\mathtt{S} + H(\mathtt{U} + \mathtt{C})$

Quantum algorithm
with complexity
$\mathtt{S} + \sqrt{H}(\mathtt{U} + \mathtt{C})$

Subroutines

S **Sample:** Sample u, according to π

U **Update:** If at u, sample v from $P[u, \cdot]$

C **Check:** If at u, check if u is marked

$P[u, v]$ = probability walker at u moves to v

π = stationary distribution of P

H = hitting time

[Szegedy 2004] [Ambainis, Gilyén, J, Kokainis 2019]

Example: Element Distinctness

Element Distinctness

Input: $x_1, \dots, x_n \in \{0, \dots, m\}$

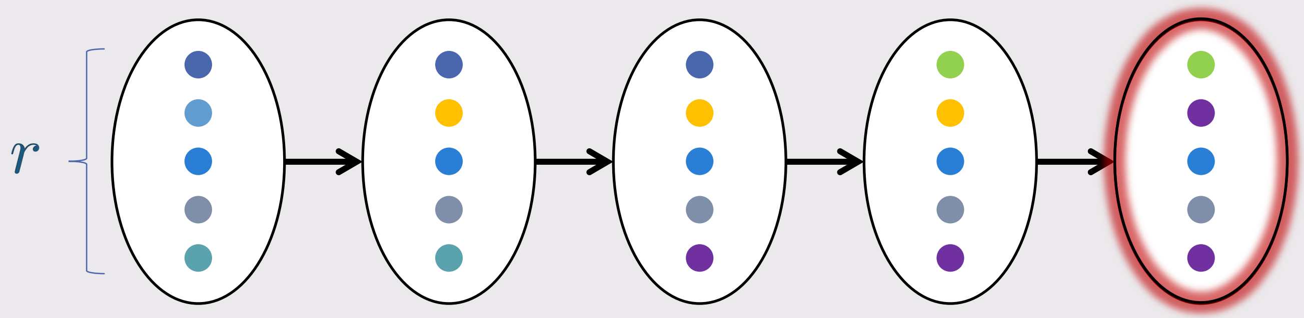
Find: a pair x_i, x_j such that $i \neq j$ and $x_i = x_j$

The Random Walk

Vertices: sets $S \subset \{x_1, \dots, x_n\}$ of size $|S| = r$

Edges: $S \sim S'$ if $|S \cap S'| = r - 1$

Marked Vertices: $\exists x_i, x_j \in S$ such that $x_i = x_j, i \neq j$



[Ambainis 2003]

Example: Element Distinctness

`Sample` $u \sim \pi$ $\quad \boxed{\mathtt{S} \approx r}$

`Repeat` H `times:` $\quad H \leq r \cdot n^2/r^2 = n^2/r$

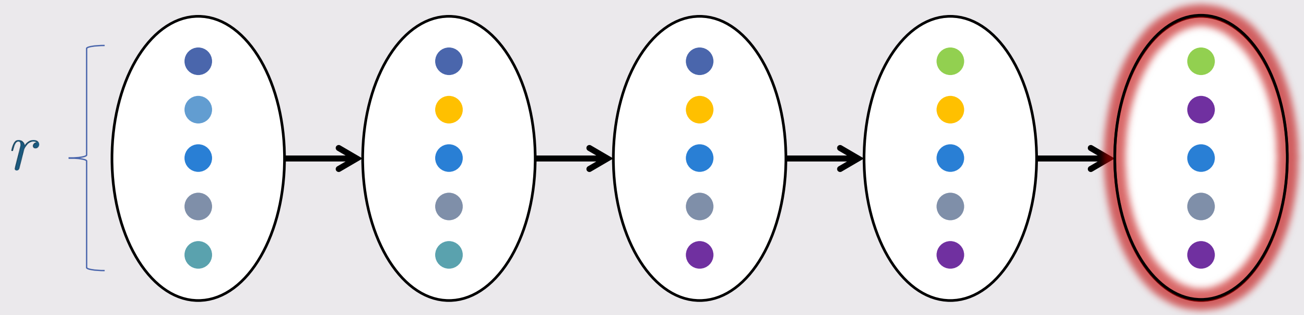
`Update` to u $\quad \boxed{\mathtt{U} \approx 1}$

`Check` u $\quad \boxed{\mathtt{C} \approx 1}$

Complexity: $\mathtt{S} + H(\mathtt{U} + \mathtt{C}) \approx r + n^2/r$

Complexity of Quantum Algorithm: $\mathtt{S} + \sqrt{H}(\mathtt{U} + \mathtt{C}) \approx r + n/\sqrt{r}$

when $r = n^{2/3}$: $\quad = n^{2/3}$



[Ambainis 2003]

Framework 1: Hitting Time Framework

Algorithmic Template

`Sample` $u \sim \pi$

`Repeat` H `times:`

  `Update to` u

  `Check` u

  `If` u `marked, output`

Complexity: $\mathtt{S} + H(\mathtt{U} + \mathtt{C})$

Quantum algorithm
with complexity
$\mathtt{S} + \sqrt{H}(\mathtt{U} + \mathtt{C})$

Subroutines

S **Sample:** Sample u, according to π

U **Update:** If at u, sample v from $P[u, \cdot]$

C **Check:** If at u, check if u is marked

$P[u, v] =$ probability walker at u moves to v

$\pi =$ stationary distribution of P

$H =$ hitting time

[Szegedy 2004] [Ambainis, Gilyén, J, Kokainis 2019]

Framework 2: Electric Network Framework

Algorithmic Template

```
Sample u ~ σ      any distribution
Repeat ??  times:
      Update to u
      Check u
      If u marked, output
```

Complexity: $\mathtt{S}_\sigma + ??(\mathtt{U} + \mathtt{C})$

Quantum algorithm
with complexity
$\mathtt{S}_\sigma + \boxed{\sqrt{C_\sigma}}(\mathtt{U} + \mathtt{C})$

???

Subroutines

$\mathtt{S}_\sigma$ **Sample:** Sample u, according to σ

$\mathtt{U}$ **Update:** If at u, sample v from $P[u,\cdot]$

$\mathtt{C}$ **Check:** If at u, check if u is marked

What is C_σ?

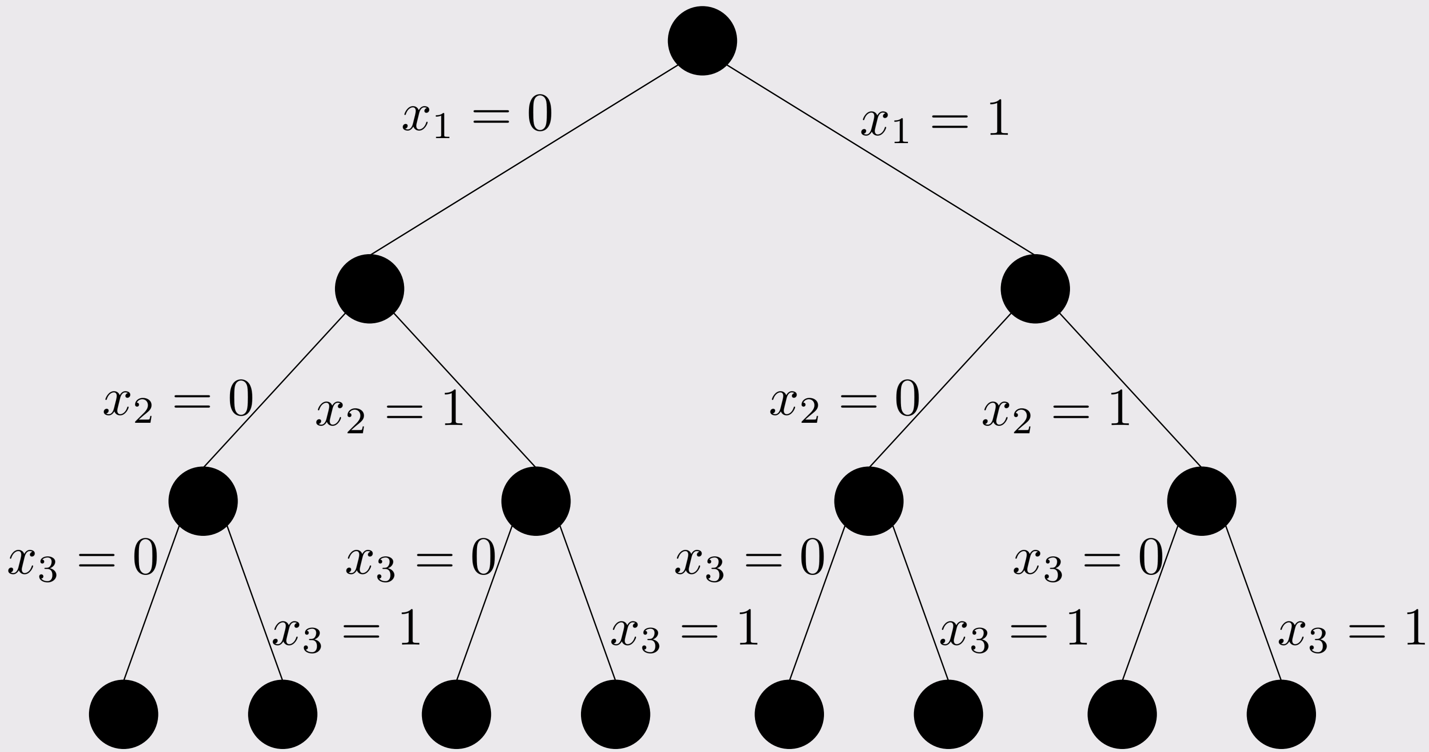
When $\sigma = \pi$: $\quad C_\pi = H$

When $\sigma(s) = 1$, $M = \{m\}$:

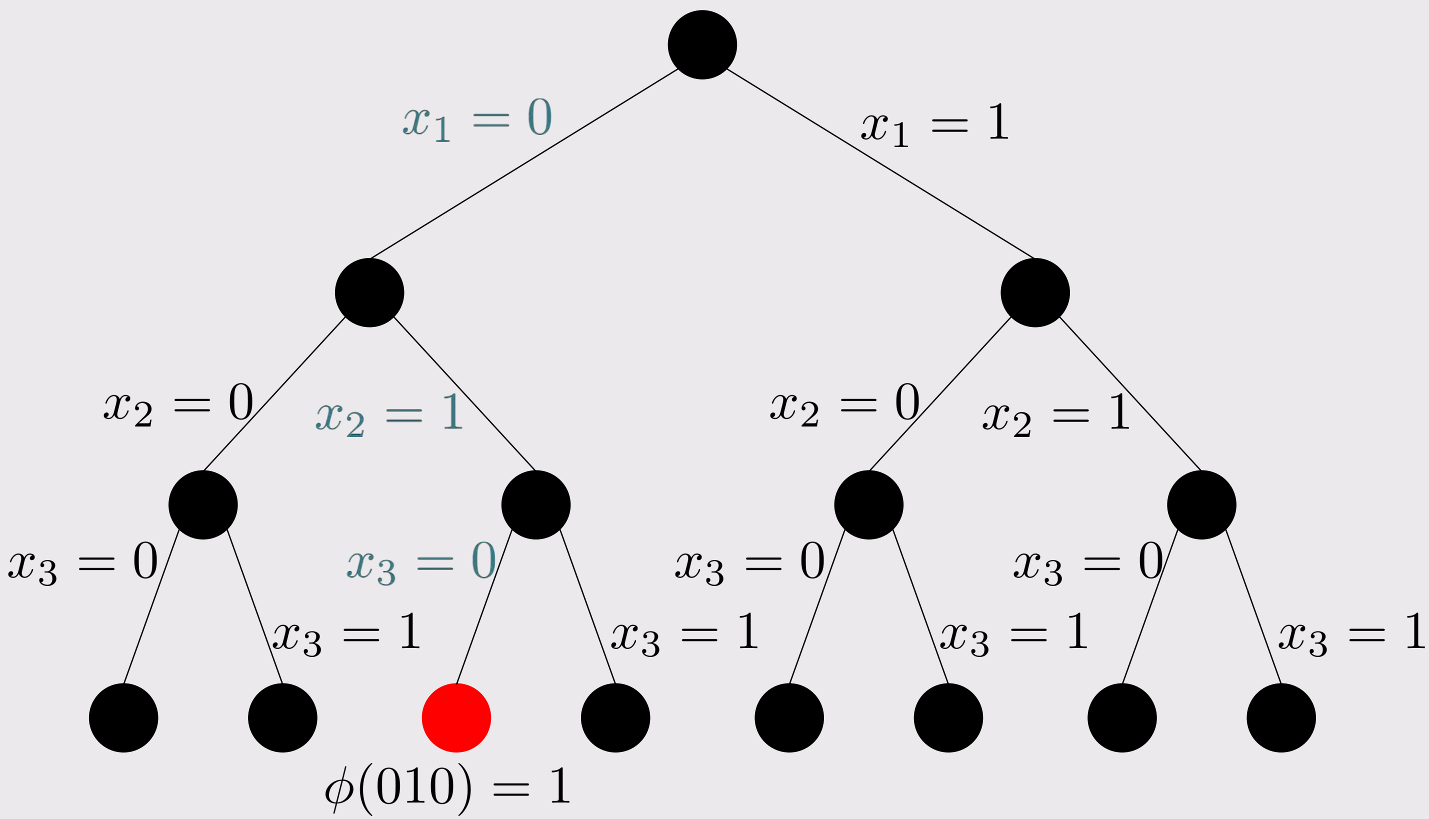
$C_\sigma =$ *commute time* from s to m

[Belovs 2013] [Apers, Gilyén, J 2019]

Example: Exploring a Search Tree



Example: Exploring a Search Tree



Example: Exploring a Search Tree

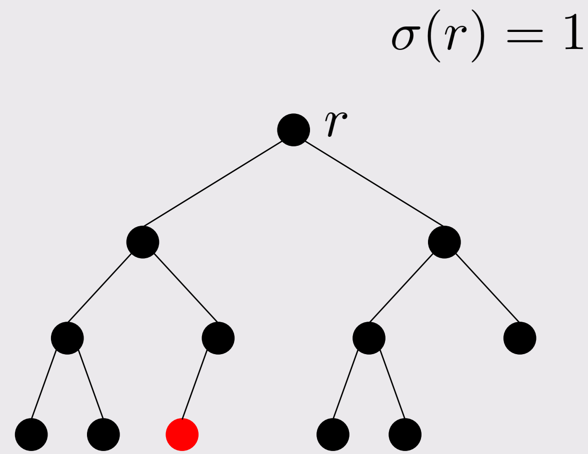
Subroutines

$\mathtt{S}_\sigma$ **Sample:** Sample u, according to σ

$\mathtt{U}$ **Update:** If at u, sample v from $P[u, \cdot]$

$\mathtt{C}$ **Check:** If at u, check if u is marked

Quantum algorithm
with complexity
$\mathtt{S}_\sigma + \sqrt{C_\sigma}(\mathtt{U} + \mathtt{C})$



[Montanaro 2015; Ambainis, Kokainis 2017]

Example: Exploring a Search Tree

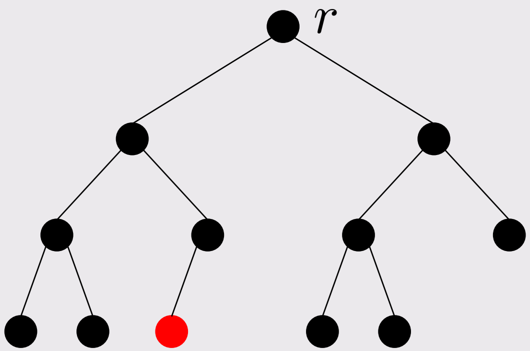
Subroutines

$\mathtt{S}_\sigma = 1$ **Sample:** Sample u, according to σ

$\mathtt{U}$ **Update:** If at u, sample v from $P[u, \cdot]$

$\mathtt{C}$ **Check:** If at u, check if u is marked

$\sigma(r) = 1$



Quantum algorithm
with complexity
$\mathtt{S}_\sigma + \sqrt{C_\sigma}(\mathtt{U} + \mathtt{C})$

[Montanaro 2015; Ambainis, Kokainis 2017]

Example: Exploring a Search Tree

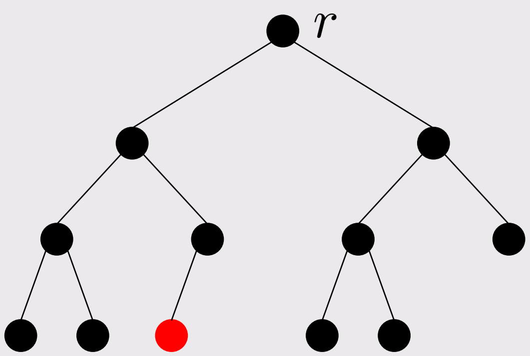
Subroutines

$\mathtt{S}_\sigma = 1$ **Sample:** Sample u, according to σ

$\mathtt{U} = 1$ **Update:** If at u, sample v from $P[u, \cdot]$

$\mathtt{C}$ **Check:** If at u, check if u is marked

$\sigma(r) = 1$



Quantum algorithm with complexity

$$\mathtt{S}_\sigma + \sqrt{C_\sigma}(\mathtt{U} + \mathtt{C})$$

[Montanaro 2015; Ambainis, Kokainis 2017]

Example: Exploring a Search Tree

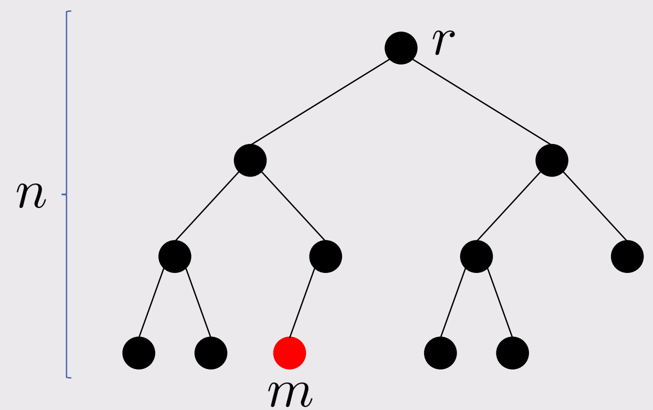
Subroutines

$\mathtt{S}_\sigma = 1$ **Sample:** Sample u, according to σ

$\mathtt{U} = 1$ **Update:** If at u, sample v from $P[u, \cdot]$

$\mathtt{C} = 1$ **Check:** If at u, check if u is marked

$\sigma(r) = 1$



Classical complexity: $\boxed{T} =$ # of nodes

$C_\sigma =$ commute time r to m

$\leq$(dist. r to m)$|E|$

$\leq nT$

Quantum algorithm with complexity

$\mathtt{S}_\sigma + \sqrt{C_\sigma}(\mathtt{U} + \mathtt{C}) \leq \boxed{\sqrt{nT}}$

[Montanaro 2015; Ambainis, Kokainis 2017]

Framework 3: MNRS Framework

Algorithmic Template

Sample $u \sim \pi$

Repeat $1/\varepsilon$ times:

$\approx$ sample π $\left\{ \begin{array}{l} \texttt{Repeat } 1/\delta \texttt{ times:} \\ \quad \texttt{Update to } u \end{array} \right.$

Check u

Complexity: $\mathtt{S} + \frac{1}{\varepsilon}(\frac{1}{\delta}\mathtt{U} + \mathtt{C})$

Quantum algorithm with complexity

$\mathtt{S} + \frac{1}{\sqrt{\varepsilon}}(\frac{1}{\sqrt{\delta}}\mathtt{U} + \mathtt{C})$

Subroutines

S **Sample:** Sample u, according to π

U **Update:** If at u, sample v from $P[u,\cdot]$

C **Check:** If at u, check if u is marked

π = stationary distribution of P

ε = probability $u \sim \pi$ marked

$1/\delta$ = mixing time of P

[Magniez, Nayak, Roland, Santha 2006]

Framework 3: MNRS Framework

Algorithmic Template

Sample $u \sim \pi$

Repeat $1/\varepsilon$ times:

$\approx$ sample π Repeat $1/\delta$ times:

Update to u

Check u

Complexity: $\mathtt{S} + \frac{1}{\varepsilon}(\frac{1}{\delta}\mathtt{U} + \mathtt{C})$

$$\frac{1}{\varepsilon} \leq H \leq \frac{1}{\varepsilon\delta}$$

Quantum algorithm with complexity

$$\mathtt{S} + \frac{1}{\sqrt{\varepsilon}}(\frac{1}{\sqrt{\delta}}\mathtt{U} + \mathtt{C})$$

Compare to Hitting Time Framework:

$$\mathtt{S} + \sqrt{H}(\mathtt{U} + \mathtt{C})$$

Example: Triangle Finding

Triangle Finding

Input: a graph G on n vertices, by its adjacency matrix A

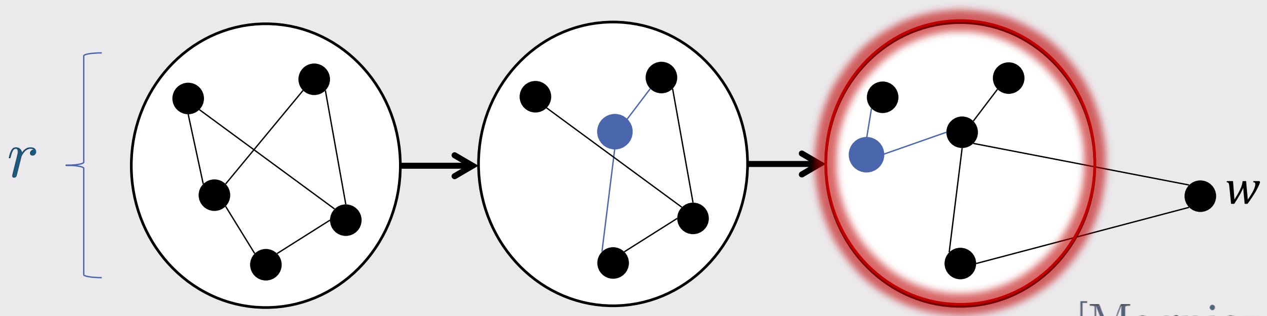
Find: a *triangle*: $u, v, w \in V(G)$ such that $A[u,v] = A[v,w] = A[w,u] = 1$

The Random Walk

Vertices: sets $S \subset V(G)$ of size $|S| = r$ and $\{uv \in E(G): u, v \in S\}$

Edges: $S \sim S'$ if $|S \cap S'| = r - 1$

Marked Vertices: $\exists u, v \in S$ such that $\exists w \in V(G)$ such that u, v, w is a triangle



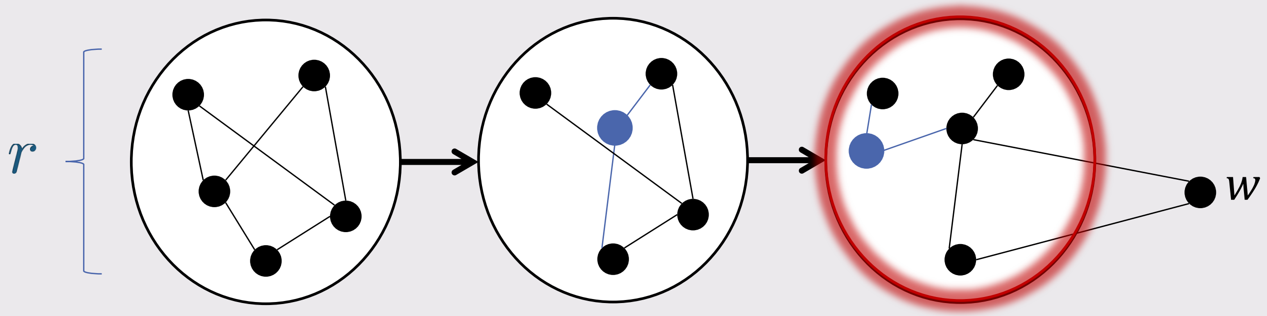
[Magniez, Santha, Szegedy 2003]

Example: Triangle Finding

Vertices: sets $S \subset V(G)$ of size $|S| = r$ and $\{uv \in E(G)\colon u, v \in S\}$

Edges: $S \sim S'$ if $|S \cap S'| = r - 1$

Marked Vertices: $\exists u, v \in S$ such that $\exists w \in V(G)$ such that u, v, w is a triangle



$\varepsilon =$ probability $u \sim \pi$ marked $\approx \frac{r^2}{n^2}$

$1/\delta =$ mixing time of $P \approx r$

$\mathtt{S} =$ sampling cost $\approx r^2$

$\mathtt{U} =$ update cost $\approx r$

$\mathtt{C} =$ checking cost

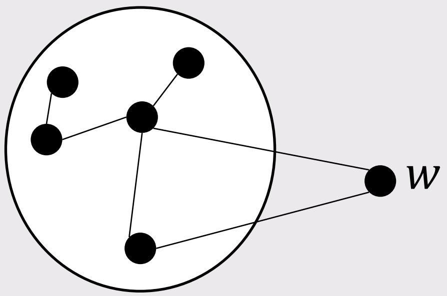
Quantum algorithm with complexity

$$\mathtt{S} + \frac{1}{\sqrt{\varepsilon}}\left(\frac{1}{\sqrt{\delta}}\mathtt{U} + \mathtt{C}\right)$$

[Magniez, Santha, Szegedy 2003]

Triangle Finding: Checking Subroutine

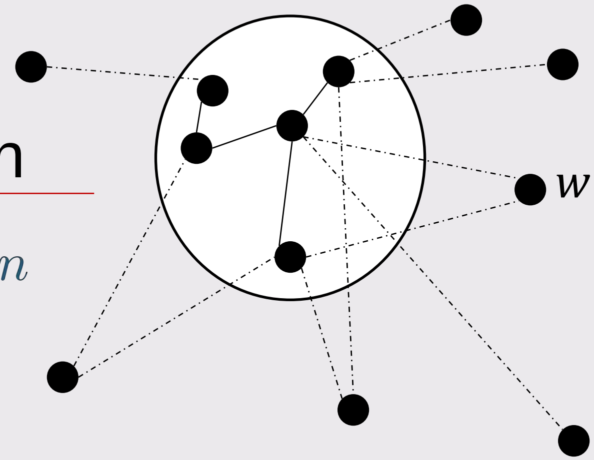
Marked Vertices: $\exists u, v \in S$ such that $\exists w \in V(G)$ such that u, v, w is a triangle



[Magniez, Santha, Szegedy 2003]

Triangle Finding: Checking Subroutine

Marked Vertices: $\exists u, v \in S$ such that $\exists w \in V(G)$ such that u, v, w is a triangle



Amplitude Amplification

Repeat $1/\varepsilon$ times: $\varepsilon \approx 1/n$

Sample w $\mathtt{S} = 0$

Check w $\mathtt{C} = r^{2/3}$

If u marked, output

Quantum algorithm with complexity

$$\frac{1}{\sqrt{\varepsilon}}(\mathtt{S} + \mathtt{C}) = \sqrt{n} r^{2/3}$$

Theorem: There is a quantum algorithm that, given an induced subgraph of G on a set S of size r, and a vertex w of G, decides if there exists an edge in S that forms a triangle with w using $r^{2/3}$ queries to G.

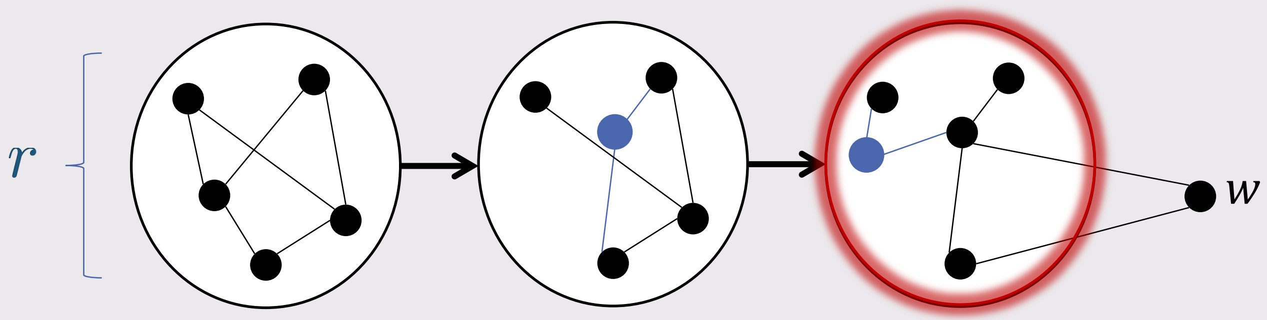
[Magniez, Santha, Szegedy 2003]

Example: Triangle Finding

Vertices: sets $S \subset V(G)$ of size $|S| = r$ and $\{uv \in E(G): u, v \in S\}$

Edges: $S \sim S'$ if $|S \cap S'| = r - 1$

Marked Vertices: $\exists u, v \in S$ such that $\exists w \in V(G)$ such that u, v, w is a triangle



$r = n^{3/5}$

$\varepsilon =$ probability $u \sim \pi$ marked $\approx \frac{r^2}{n^2} = \frac{1}{n^{4/5}}$

$1/\delta =$ mixing time of $P \approx r = n^{3/5}$

$\mathtt{S} =$ sampling cost $\approx r^2 = n^{6/5}$

$\mathtt{U} =$ update cost $\approx r = n^{3/5}$

$\mathtt{C} =$ checking cost $= \sqrt{n} r^{2/3} = n^{9/10}$

Quantum algorithm with complexity

$$\mathtt{S} + \frac{1}{\sqrt{\varepsilon}}\left(\frac{1}{\sqrt{\delta}}\mathtt{U} + \mathtt{C}\right)$$

$$= n^{13/10}$$

[Magniez, Santha, Szegedy 2003]

Framework 3: MNRS Framework

Algorithmic Template

Sample $u \sim \pi$

Repeat $1/\varepsilon$ times:

$\approx$ sample π — Repeat $1/\delta$ times:

Update to u

Check u

Complexity: $\mathtt{S} + \frac{1}{\varepsilon}(\frac{1}{\delta}\mathtt{U} + \mathtt{C})$

$\frac{1}{\varepsilon} \leq H \leq \frac{1}{\varepsilon\delta}$

Quantum algorithm with complexity

$\mathtt{S} + \frac{1}{\sqrt{\varepsilon}}(\frac{1}{\sqrt{\delta}}\mathtt{U} + \mathtt{C})$

Compare to Hitting Time Framework:

$\mathtt{S} + \sqrt{H}(\mathtt{U} + \mathtt{C})$

Best of Both Worlds?

MNRS Framework

$$\mathtt{S} + \frac{1}{\sqrt{\varepsilon}}\left(\frac{1}{\sqrt{\delta}}\mathtt{U} + \mathtt{C}\right)$$

optimal number of checks

Hitting Time Framework:

$$\mathtt{S} + \sqrt{H}(\mathtt{U} + \mathtt{C})$$

optimal number of updates

$$\frac{1}{\varepsilon} \leq H \leq \frac{1}{\varepsilon\delta}$$

Assuming only one marked vertex,
Controlled Quantum Amplification:

$$\mathtt{S} + \sqrt{H}\mathtt{U} + \frac{1}{\sqrt{\varepsilon}}\mathtt{C}$$

[Dohotaru, Høyer 2017]

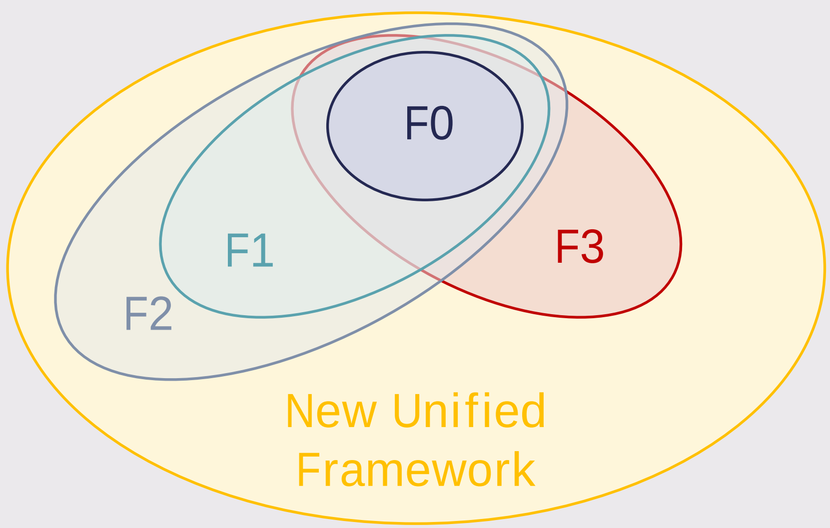
Different Frameworks

F0: Amplitude Amplification $O(\frac{1}{\sqrt{\varepsilon}}(\mathtt{S}+\mathtt{C}))$

F1: Hitting Time Framework $O(\mathtt{S}+\sqrt{H}(\mathtt{U}+\mathtt{C}))$

F2: Electric Network Framework $O(\mathtt{S}(\sigma)+\sqrt{C_\sigma}(\mathtt{U}+\mathtt{C}))$

F3: MNRS Framework $O(\mathtt{S}+\frac{1}{\sqrt{\varepsilon}}(\frac{1}{\sqrt{\delta}}\mathtt{U}+\mathtt{C}))$



Subroutines

- $\mathtt{S}$: sample
- $\mathtt{U}$: update
- $\mathtt{C}$: check

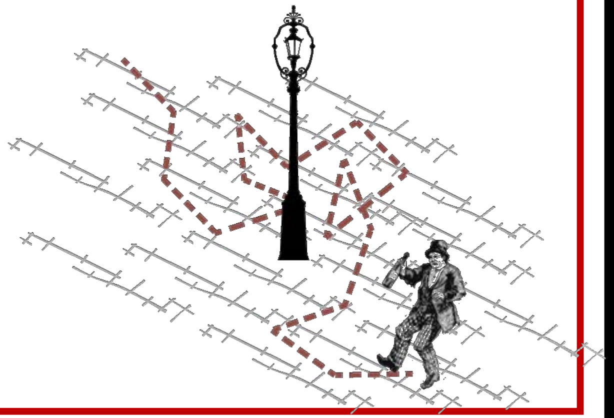
Parameters

$$H = \text{hitting time}$$
$$1/\delta \approx \text{mixing time}$$
$$\varepsilon = \textstyle\sum_{m\in M}\pi(m)$$

For Comparison

$$\frac{1}{\varepsilon} \le H \le \frac{1}{\varepsilon\delta}$$
$$C_\pi = H$$

Unified Framework



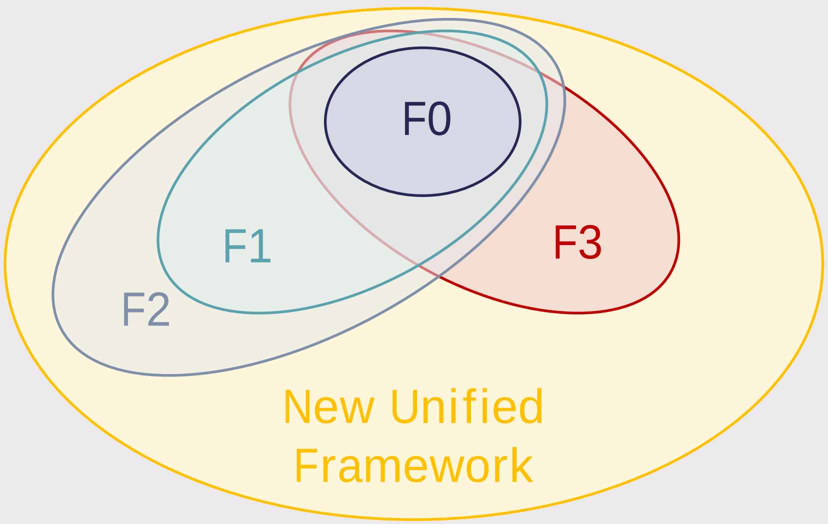
Different Frameworks

F0: Amplitude Amplification $O(\frac{1}{\sqrt{\varepsilon}}(\mathtt{S}+\mathtt{C}))$

F1: Hitting Time Framework $O(\mathtt{S}+\sqrt{H}(\mathtt{U}+\mathtt{C}))$

F2: Electric Network Framework $O(\mathtt{S}(\sigma)+\sqrt{C_\sigma}(\mathtt{U}+\mathtt{C}))$

F3: MNRS Framework $O(\mathtt{S}+\frac{1}{\sqrt{\varepsilon}}(\frac{1}{\sqrt{\delta}}\mathtt{U}+\mathtt{C}))$



Subroutines

$\mathtt{S}$: sample

$\mathtt{U}$: update

$\mathtt{C}$: check

Parameters

$$H = \text{hitting time}$$
$$1/\delta \approx \text{mixing time}$$
$$\varepsilon = \sum_{m\in M}\pi(m)$$

For Comparison

$$\frac{1}{\varepsilon} \le H \le \frac{1}{\varepsilon\delta}$$
$$C_\pi = H$$

Unified Framework

MNRS Framework

```
Sample u ~ π
Repeat 1/ε times:
        Repeat 1/δ times:
              Update to u
        Check u
```

Subroutines

Sample: Sample u, according to π

Update: If at u, sample v from $P[u, \cdot]$

Check: If at u, check if u is marked

Unified Framework

Algorithmic Template

```
Sample u ~ π
Repeat ?? times:
    Repeat t times:
        Update to u
    Check u
```

One step of P^t

Subroutines

Sample: Sample u, according to π

Update: If at u, sample v from $P[u, \cdot]$

Check: If at u, check if u is marked

t-step walk P^t

P^t: each step, take t steps of P

$t =$ free parameter

[Apers, Gilyén, J 2019]

Unified Framework

Algorithmic Template

Sample $u \sim \pi$

Repeat $H(P^t)$ times:

One step of P^t

Repeat t times:

Update to u

Check u

Complexity: $\mathtt{S} + H(P^t)(t\mathtt{U} + \mathtt{C})$

Quantum algorithm with complexity

$\mathtt{S} + \sqrt{H(P^t)}(\sqrt{t}\mathtt{U} + \mathtt{C})$

Subroutines

S **Sample:** Sample u, according to π

U **Update:** If at u, sample v from $P[u, \cdot]$

C **Check:** If at u, check if u is marked

let $t = 1/\delta$

$H(P^{1/\delta}) \approx 1/\varepsilon$

$$\underbrace{\mathtt{S} + \frac{1}{\sqrt{\varepsilon}}\left(\frac{1}{\sqrt{\delta}}\mathtt{U} + \mathtt{C}\right)}_{\text{MNRS}}$$

= free parameter

$H(P^t)$ = hitting time of P^t

Final Unified Framework

Algorithmic Template

`Sample` $u \sim \sigma$

`Repeat` $??$ `times:`

One step of P^t

`Repeat` t `times:`

`Update to` u

`Check` u

Complexity: $\mathtt{S}_\sigma + ??(t\mathtt{U} + \mathtt{C})$

Quantum algorithm with complexity

$$\mathtt{S}_\sigma + \sqrt{C_\sigma(P^t)}(\sqrt{t}\mathtt{U} + \mathtt{C})$$

Subroutines

$\mathtt{S}_\sigma$ **Sample:** Sample u, according to σ

$\mathtt{U}$ **Update:** If at u, sample v from $P[u,\cdot]$

$\mathtt{C}$ **Check:** If at u, check if u is marked

[Apers, Gilyén, J 2019]

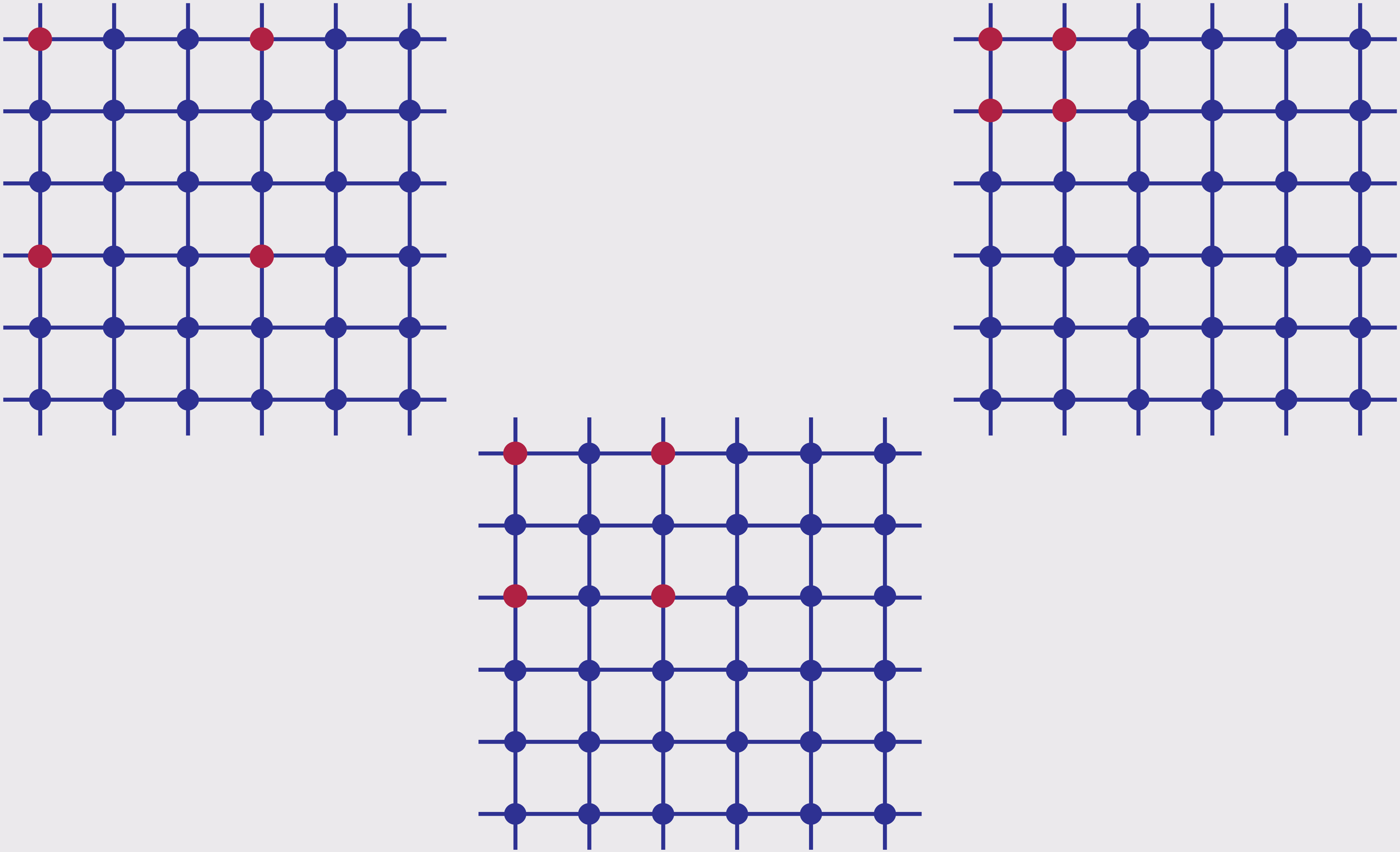
Final Unified Framework

Unified Framework	$O(\mathtt{S}_\sigma + \sqrt{C_\sigma(P^t)}(\sqrt{t}\mathtt{U} + \mathtt{C}))$
F1: Hitting Time Framework	$\sigma = \pi,\ t = 1$
F2: Electric Network Framework	any σ, $t = 1$
F3: MNRS Framework	$\sigma = \pi,\ t = 1/\delta$

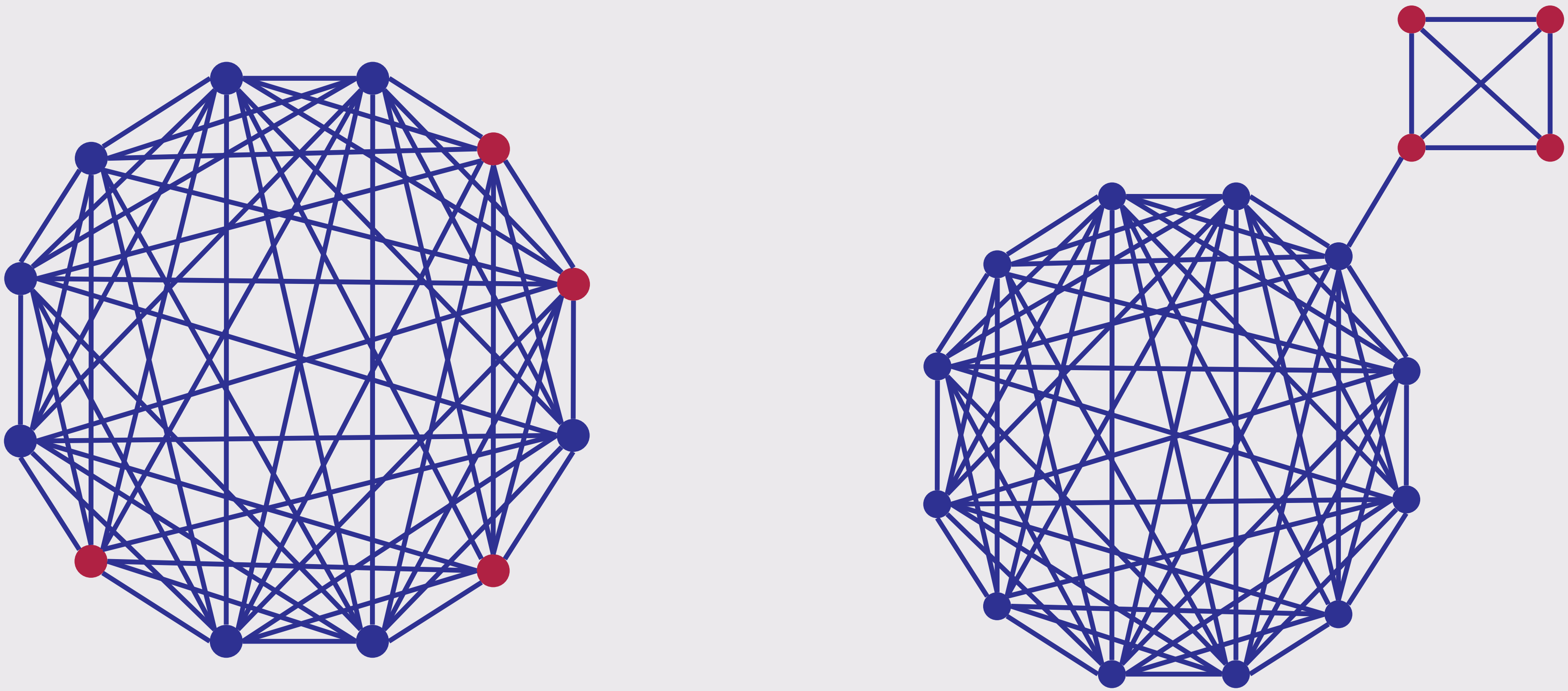
$\sigma = \pi$, $t = \varepsilon H$, one marked vertex $\quad O(\mathtt{S} + \sqrt{H}\mathtt{U} + \frac{1}{\sqrt{\varepsilon}}\mathtt{C})$

[Apers, Gilyén, J 2019]

Checking Frequency



Checking Frequency



Final Unified Framework

Unified Framework	$O(\mathtt{S}_\sigma + \sqrt{C_\sigma(P^t)}(\sqrt{t}\mathtt{U} + \mathtt{C}))$
F1: Hitting Time Framework	$\sigma = \pi,\ t = 1$
F2: Electric Network Framework	any σ, $t = 1$
F3: MNRS Framework	$\sigma = \pi,\ t = 1/\delta$

[Apers, Gilyén, J 2019]