Simple dynamic algorithms for Maximal Independent Set, Maximum Flow and Maximum Matching

Manoj Gupta, Shahbaz Khan

Reading Group Algorithms

Antonis Skarlatos

PARIS LODRON UNIVERSITÄT SALZBURG

Outline of the paper

- Maximal Independent Set (MIS)
  - Fully dynamic under edge updates
  - $O(\min\{\Delta, m^{\frac{2}{3}}\})$ amortized update time
- Unit-Capacity Maximum Flow (UMF)
  - Incremental under edge updates
  - $O(n)$ amortized update time
- Maximum Cardinality Matching (MCM)
  - Incremental under edge updates
  - $O(n)$ amortized update time

Outline of the paper

- Maximal Independent Set (MIS)
  - Fully dynamic under edge updates
  - $O(\min\{\Delta, m^{\frac{2}{3}}\})$ amortized update time
- Unit-Capacity Maximum Flow (UMF)
  - Incremental under edge updates
  - $O(n)$ amortized update time
- Maximum Cardinality Matching (MCM)
  - Incremental under edge updates
  - $O(n)$ amortized update time

Outline of the paper

- Maximal Independent Set (MIS)
  - Fully dynamic under edge updates
  - $O(\min\{\Delta, m^{\frac{2}{3}}\})$ amortized update time
- Unit-Capacity Maximum Flow (UMF)
  - Incremental under edge updates
  - $O(n)$ amortized update time
- Maximum Cardinality Matching (MCM)
  - Incremental under edge updates
  - $O(n)$ amortized update time

Maximal Independent Set (MIS)

Definition of MIS

$G = (V, E)$. A subset $M \subseteq V$ is called a *maximal independent set* if and only if:

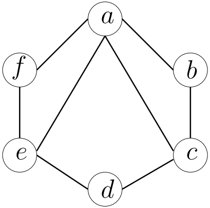
1. $\forall u, w \in M$: no edge between them
2. $(\forall u \notin M)$ $(\exists w \in M)$: edge between them

Maximal Independent Set (MIS)

Definition of MIS

$G = (V, E)$. A subset $M \subseteq V$ is called a *maximal independent set* if and only if:

1. $\forall u, w \in M$: no edge between them
2. $(\forall u \notin M)$ $(\exists w \in M)$: edge between them



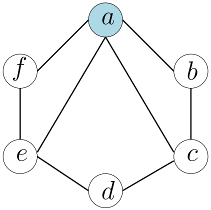
$M = \emptyset$

Maximal Independent Set (MIS)

Definition of MIS

$G = (V, E)$. A subset $M \subseteq V$ is called a *maximal independent set* if and only if:

1. $\forall u, w \in M$: no edge between them
2. $(\forall u \notin M)\ (\exists w \in M)$: edge between them



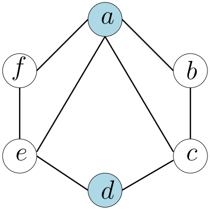
$M = \{a\}$

Maximal Independent Set (MIS)

Definition of MIS

$G = (V, E)$. A subset $M \subseteq V$ is called a *maximal independent set* if and only if:

1. $\forall u, w \in M$: no edge between them
2. $(\forall u \notin M)$ $(\exists w \in M)$: edge between them



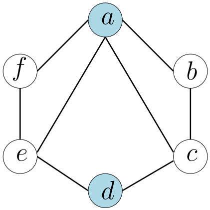
$M = \{a, d\}$

Maximal Independent Set (MIS)

Definition of MIS

$G = (V, E)$. A subset $M \subseteq V$ is called a *maximal independent set* if and only if:

1. $\forall u, w \in M$: no edge between them
2. $(\forall u \notin M)$ $(\exists w \in M)$: edge between them



$M = \{a, d\}$

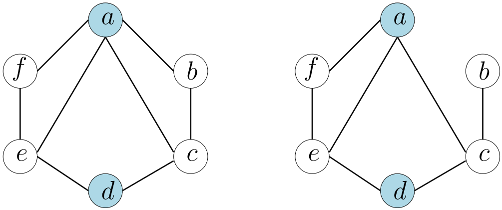
DFS/BFS: $O(n + m)$

Maximal Independent Set (MIS)

Definition of MIS

$G = (V, E)$. A subset $M \subseteq V$ is called a *maximal independent set* if and only if:

1. $\forall u, w \in M$: no edge between them
2. $(\forall u \notin M)\ (\exists w \in M)$: edge between them

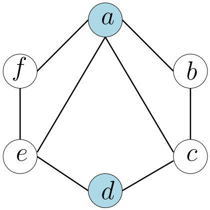


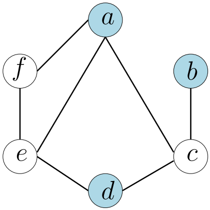
Maximal Independent Set (MIS)

Definition of MIS

$G = (V, E)$. A subset $M \subseteq V$ is called a *maximal independent set* if and only if:

1. $\forall u, w \in M$: no edge between them
2. $(\forall u \notin M)\ (\exists w \in M)$: edge between them



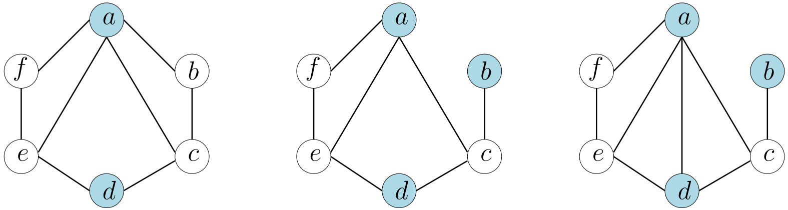


Maximal Independent Set (MIS)

Definition of MIS

$G = (V, E)$. A subset $M \subseteq V$ is called a *maximal independent set* if and only if:

1. $\forall u, w \in M$: no edge between them
2. $(\forall u \notin M)$ $(\exists w \in M)$: edge between them

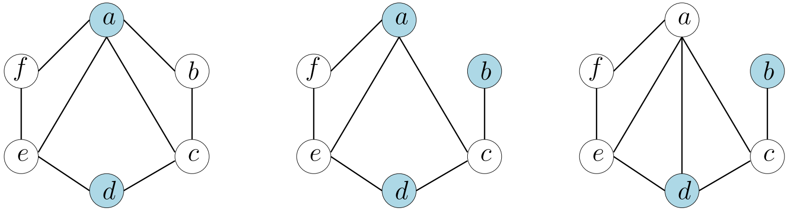


Maximal Independent Set (MIS)

Definition of MIS

$G = (V, E)$. A subset $M \subseteq V$ is called a *maximal independent set* if and only if:

1. $\forall u, w \in M$: no edge between them
2. $(\forall u \notin M)$ $(\exists w \in M)$: edge between them

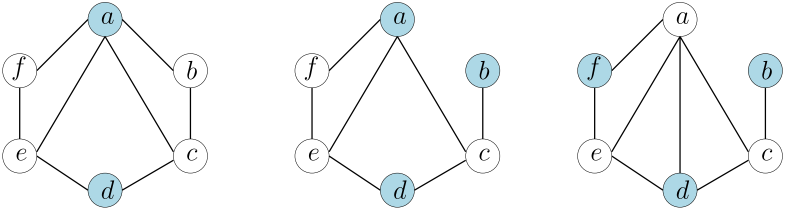


Maximal Independent Set (MIS)

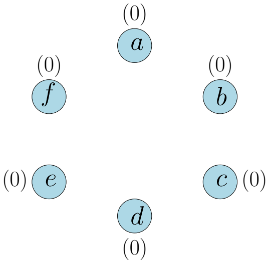
Definition of MIS

$G = (V, E)$. A subset $M \subseteq V$ is called a *maximal independent set* if and only if:

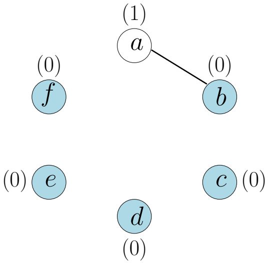
1. $\forall u, w \in M$: no edge between them
2. $(\forall u \notin M)$ $(\exists w \in M)$: edge between them



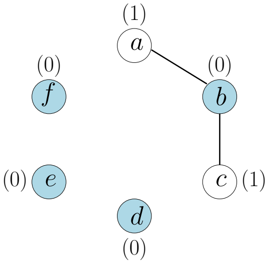
Algorithm with counters



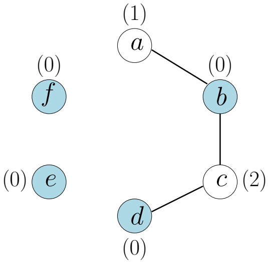
Algorithm with counters



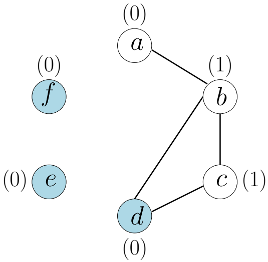
Algorithm with counters



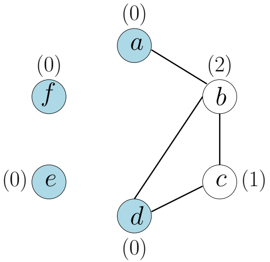
Algorithm with counters



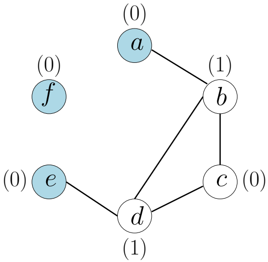
Algorithm with counters



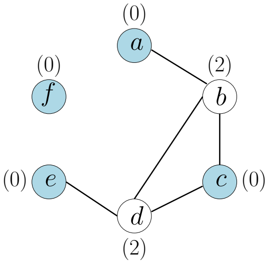
Algorithm with counters



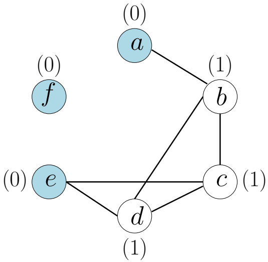
Algorithm with counters



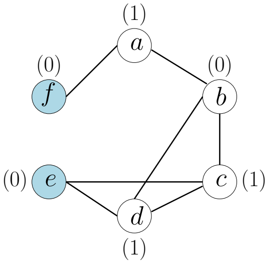
Algorithm with counters



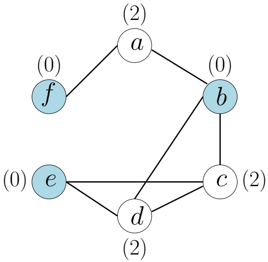
Algorithm with counters



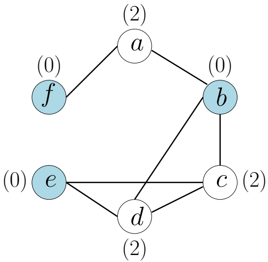
Algorithm with counters



Algorithm with counters



Algorithm with counters

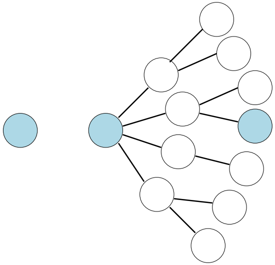


Edge deletion: one vertex may be added to MIS

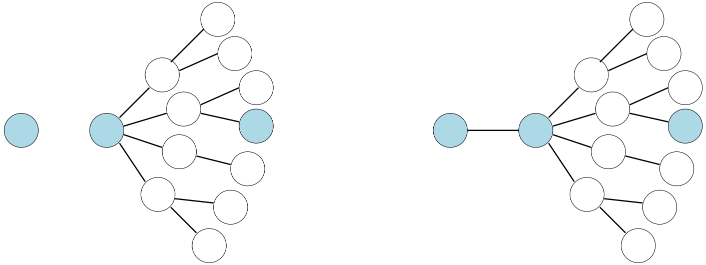
Edge insertion: one vertex may be removed from MIS

multiple vertices may be added to MIS

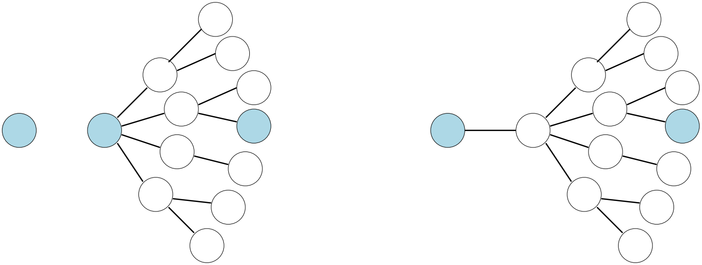
Difficult Case



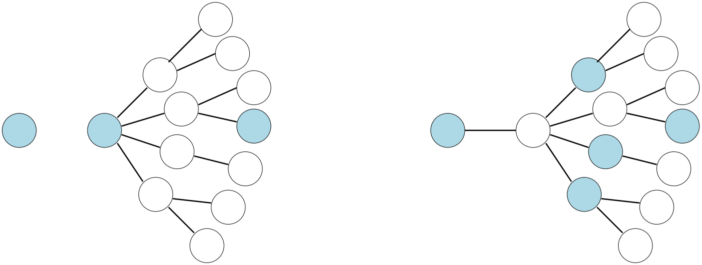
Difficult Case



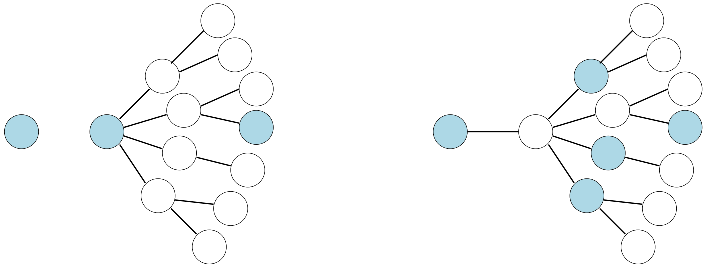
Difficult Case



Difficult Case

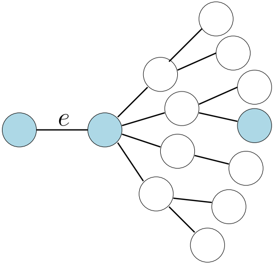


Difficult Case

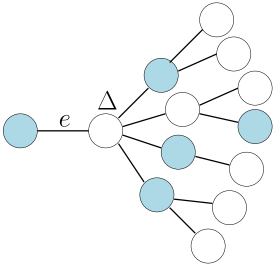


$O(\Delta^2)$ worst-case time

$O(\Delta)$ amortized update time

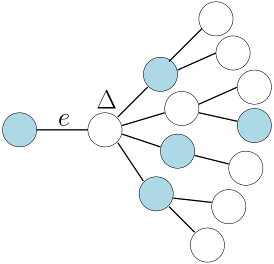


$O(\Delta)$ amortized update time



1. At most one vertex may be removed from MIS
2. A vertex iterates over its neighborhood only if enters/leaves MIS

$O(\Delta)$ amortized update time

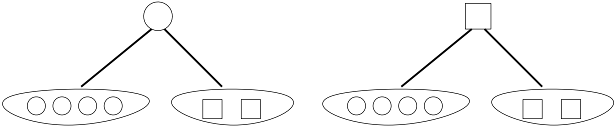


e_1	e_2	e_3	e_4	e_5
-a	-b	-c	-a	$+a$
		$+a$	$+c$	
		+b		

1. At most one vertex may be removed from MIS
2. A vertex iterates over its neighborhood only if enters/leaves MIS

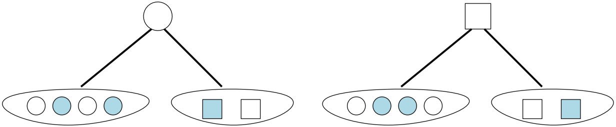
Heavy/Light separation

$$L = \{u \in V(G) \mid \deg(u) \leq \alpha\}$$
$$H = \{u \in V(G) \mid \deg(u) > \alpha\}$$



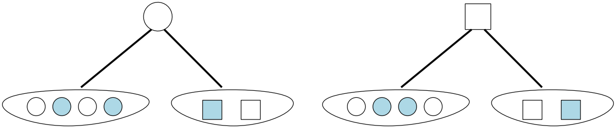
Heavy/Light separation

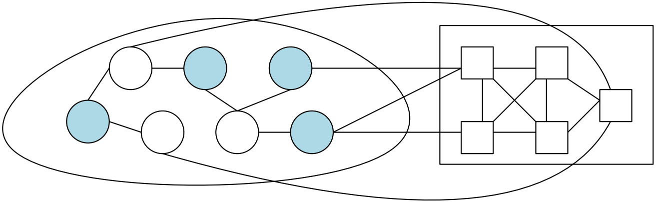
$$L = \{u \in V(G) \mid \deg(u) \leq \alpha\}$$
$$H = \{u \in V(G) \mid \deg(u) > \alpha\}$$



Heavy/Light separation

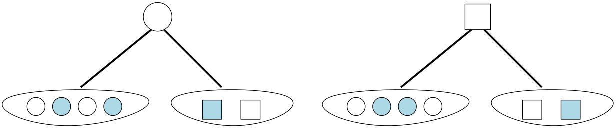
$$L = \{u \in V(G) \mid \deg(u) \leq \alpha\}$$
$$H = \{u \in V(G) \mid \deg(u) > \alpha\}$$

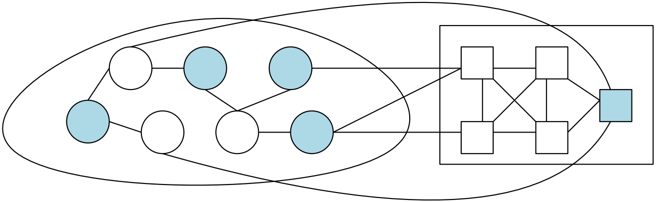




Heavy/Light separation

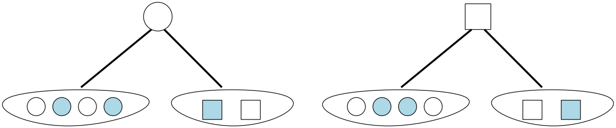
$$L = \{u \in V(G) \mid \deg(u) \leq \alpha\}$$
$$H = \{u \in V(G) \mid \deg(u) > \alpha\}$$

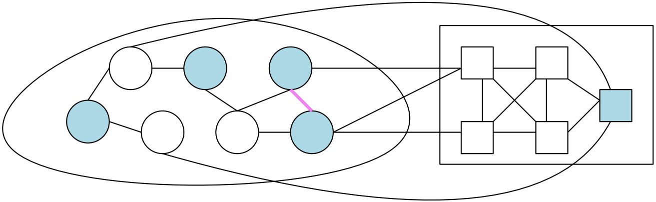




Heavy/Light separation

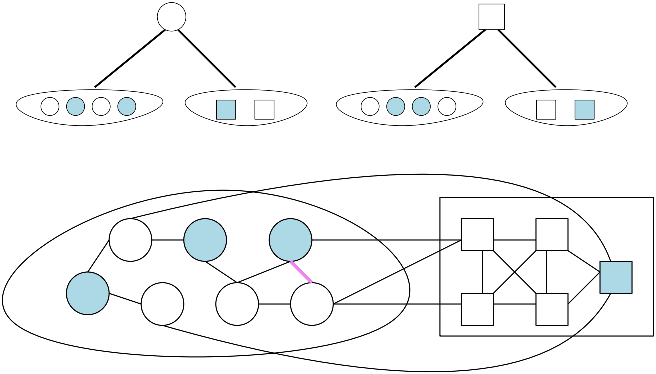
$$L = \{u \in V(G) \mid \deg(u) \leq \alpha\}$$
$$H = \{u \in V(G) \mid \deg(u) > \alpha\}$$





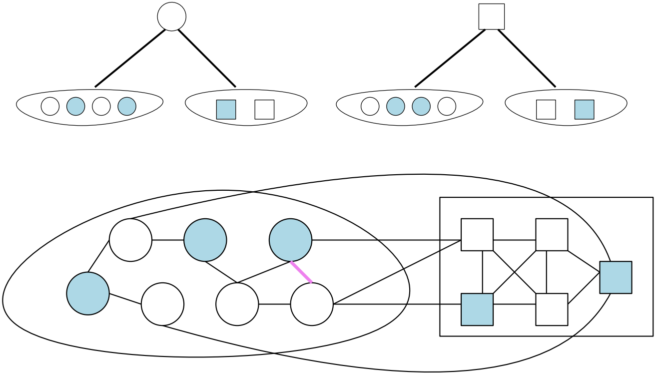
Heavy/Light separation

$$L = \{u \in V(G) \mid \deg(u) \leq \alpha\}$$
$$H = \{u \in V(G) \mid \deg(u) > \alpha\}$$



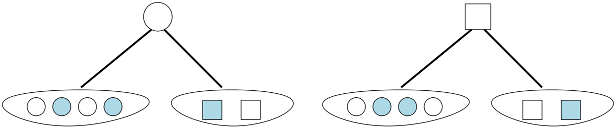
Heavy/Light separation

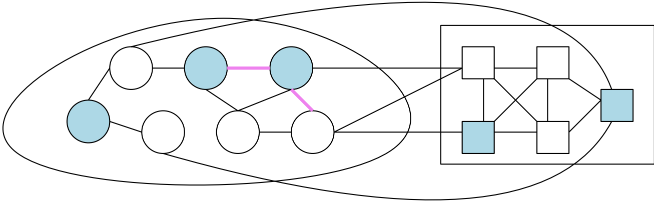
$$L = \{u \in V(G) \mid \deg(u) \leq \alpha\}$$
$$H = \{u \in V(G) \mid \deg(u) > \alpha\}$$



Heavy/Light separation

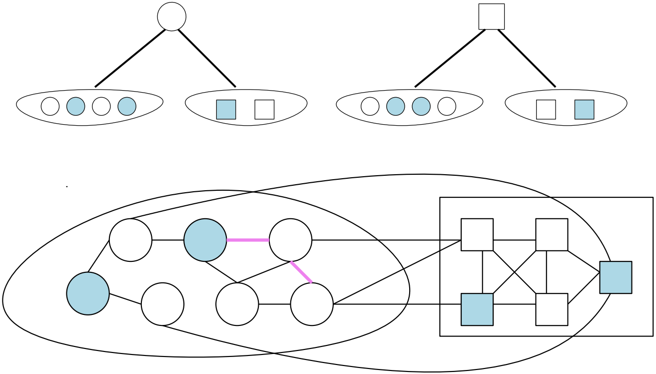
$$L = \{u \in V(G) \mid \deg(u) \leq \alpha\}$$
$$H = \{u \in V(G) \mid \deg(u) > \alpha\}$$





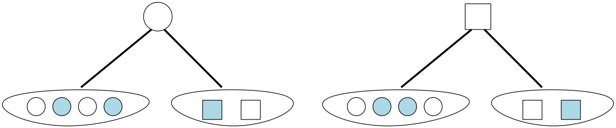
Heavy/Light separation

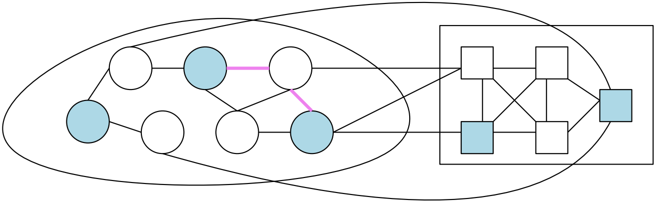
$$L = \{u \in V(G) \mid \deg(u) \leq \alpha\}$$
$$H = \{u \in V(G) \mid \deg(u) > \alpha\}$$



Heavy/Light separation

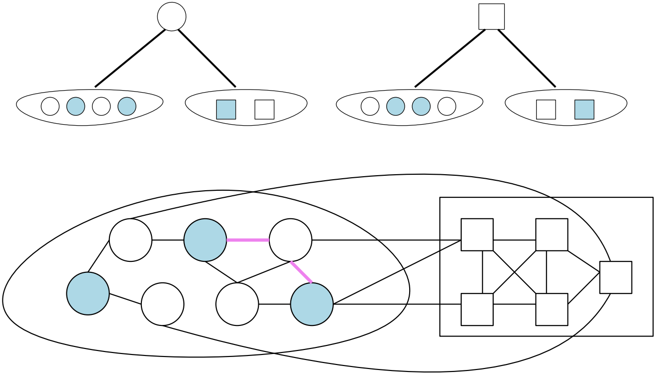
$$L = \{u \in V(G) \mid \deg(u) \leq \alpha\}$$
$$H = \{u \in V(G) \mid \deg(u) > \alpha\}$$





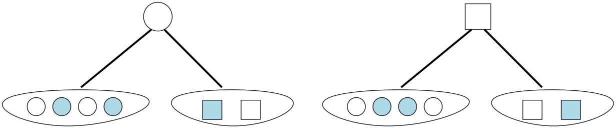
Heavy/Light separation

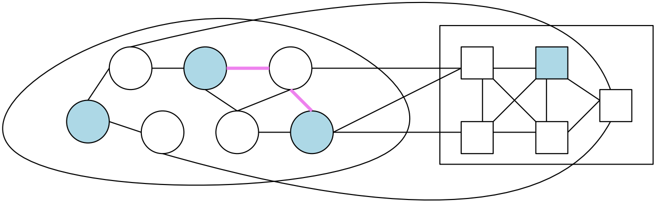
$$L = \{u \in V(G) \mid \deg(u) \leq \alpha\}$$
$$H = \{u \in V(G) \mid \deg(u) > \alpha\}$$



Heavy/Light separation

$$L = \{u \in V(G) \mid \deg(u) \leq \alpha\}$$
$$H = \{u \in V(G) \mid \deg(u) > \alpha\}$$





Analyzing Time Complexity

- $O(\alpha) + O(|E(H)|)$

Analyzing Time Complexity

- $O(\alpha) + O(|E(H)|)$
- $|H| \leq O(\frac{m}{\alpha}) \Rightarrow |E(H)| \leq O((\frac{m}{\alpha})^2)$

Analyzing Time Complexity

- $O(\alpha) + O(|E(H)|)$
- $|H| \leq O(\frac{m}{\alpha}) \Rightarrow |E(H)| \leq O((\frac{m}{\alpha})^2)$
- $\alpha = (\frac{m}{\alpha})^2 \Rightarrow \alpha = m^{\frac{2}{3}}$

Analyzing Time Complexity

- $O(\alpha) + O(|E(H)|)$
- $|H| \leq O(\frac{m}{\alpha}) \Rightarrow |E(H)| \leq O((\frac{m}{\alpha})^2)$
- $\alpha = (\frac{m}{\alpha})^2 \Rightarrow \alpha = m^{\frac{2}{3}}$

Deterministic Algorithm:

$O(\min\{\Delta, m^{\frac{2}{3}}\})$ amortized update time

$O(1)$ worst-case query time

Analyzing Time Complexity

- $O(\alpha) + O(|E(H)|)$
- $|H| \leq O(\frac{m}{\alpha}) \Rightarrow |E(H)| \leq O((\frac{m}{\alpha})^2)$
- $\alpha = (\frac{m}{\alpha})^2 \Rightarrow \alpha = m^{\frac{2}{3}}$

Deterministic Algorithm:

$O(\min\{\Delta, m^{\frac{2}{3}}\})$ amortized update time

$O(1)$ worst-case query time

Why is this algorithm faster?

Analyzing Time Complexity

- $O(\alpha) + O(|E(H)|)$
- $|H| \leq O(\frac{m}{\alpha}) \;\Rightarrow\; |E(H)| \leq O((\frac{m}{\alpha})^2)$
- $\alpha = (\frac{m}{\alpha})^2 \;\Rightarrow\; \alpha = m^{\frac{2}{3}}$

Deterministic Algorithm:

$O(\min\{\Delta, m^{\frac{2}{3}}\})$ amortized update time

$O(1)$ worst-case query time

Why is this algorithm faster?

Heavy vertices do not inform their light neighborhood

Summary of the algorithm

- Split the vertices into two groups: light and heavy

Summary of the algorithm

- Split the vertices into two groups: light and heavy
- First step: Apply the $O(\Delta)$ algorithm for the light vertices

Summary of the algorithm

- ▷ Split the vertices into two groups: light and heavy
- ▷ First step: Apply the $O(\Delta)$ algorithm for the light vertices
- ▷ Second step: Apply the $O(m)$ algorithm for the heavy vertices with no light neighbor in MIS

Summary of the algorithm

- ▷ Split the vertices into two groups: light and heavy
- ▷ First step: Apply the $O(\Delta)$ algorithm for the light vertices
- ▷ Second step: Apply the $O(m)$ algorithm for the heavy vertices with no light neighbor in MIS

Thank you !

[1] Assadi et. al. "Fully dynamic maximal independent set with sublinear in n update time". **in** 2019.

[2] Manoj Gupta **and** Shahbaz Khan. "Simple dynamic algorithms for Maximal Independent Set, Maximum Flow and Maximum Matching". **in** 2021.