Fully Dynamic Maximal Independent Set in Expected Poly-Log Update Time

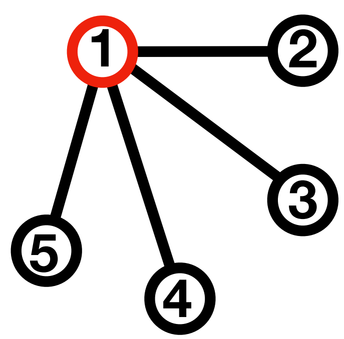
Shiri Chechik, Tianyi Zhang

Tel Aviv University, Tsinghua University

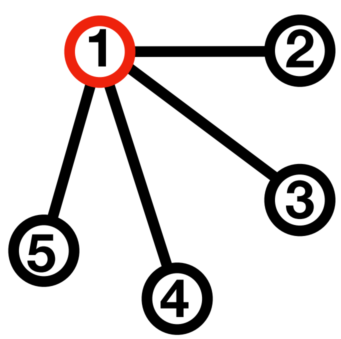
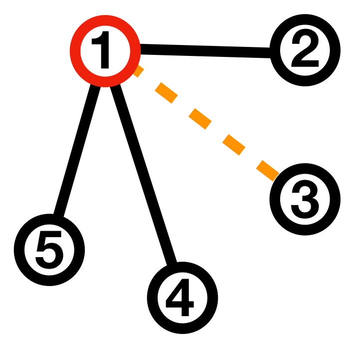
Definition: dynamic MIS

- Given an undirected simple graph G = (V, E)
- A sequence of edge ins / del by an oblivious adversary
- Explicitly maintain a maximal independent set

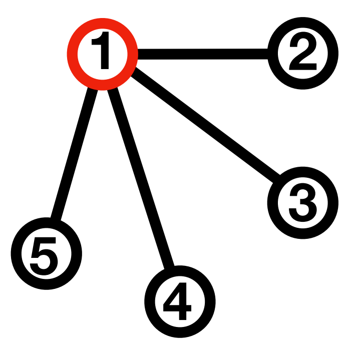
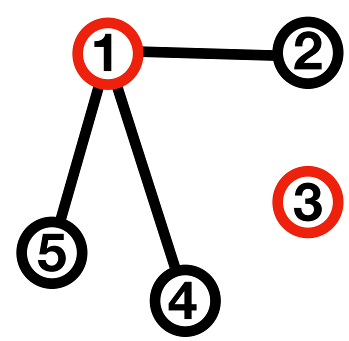
<u>Example:</u> dynamic MIS

Input: Updates to G				
Picture				
Output: Changes to the MIS				

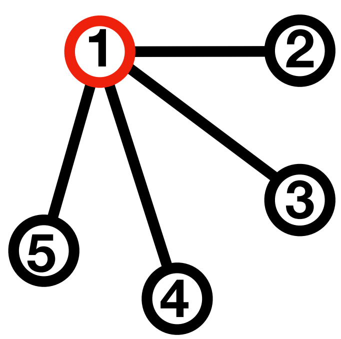
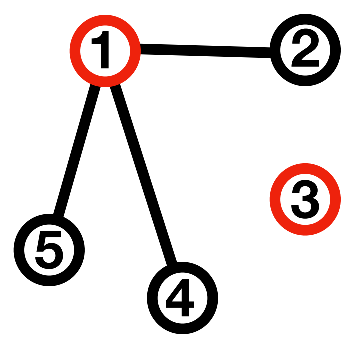
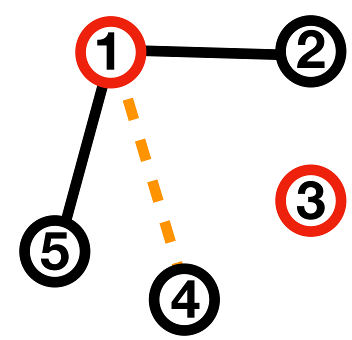
Example: dynamic MIS

Input: Updates to G		Delete (1, 3)		
Picture				
Output: Changes to the MIS				

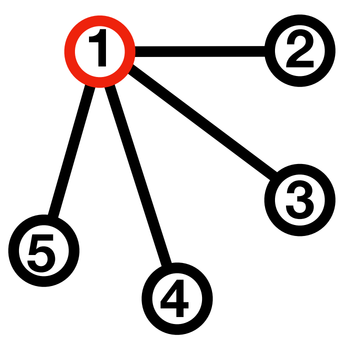
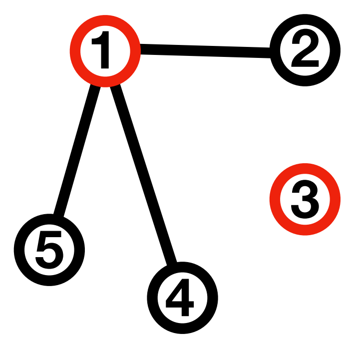
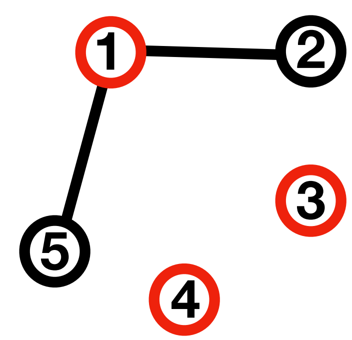
Example: dynamic MIS

Input: Updates to G		Delete (1, 3)		
Picture				
Output: Changes to the MIS		3 joins MIS		

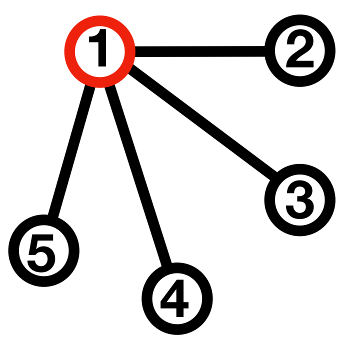
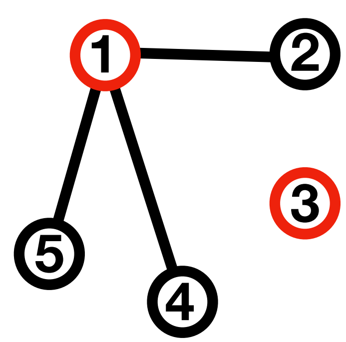
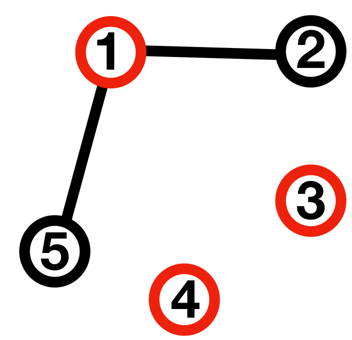
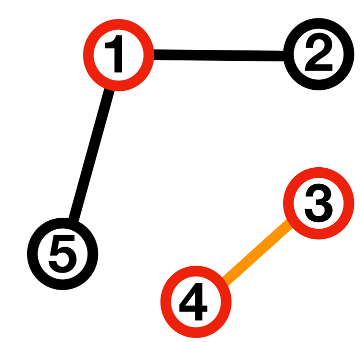
<u>Example:</u> dynamic MIS

Input: Updates to G		Delete (1, 3)	Delete (1, 4)	
Picture				
Output: Changes to the MIS		3 joins MIS		

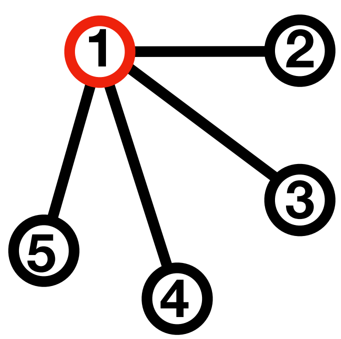
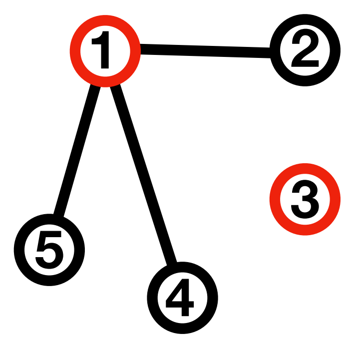
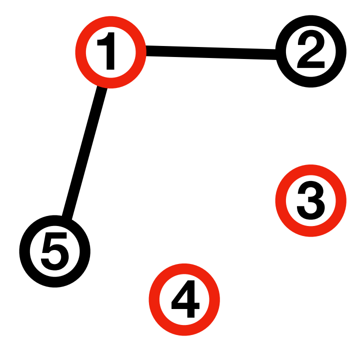
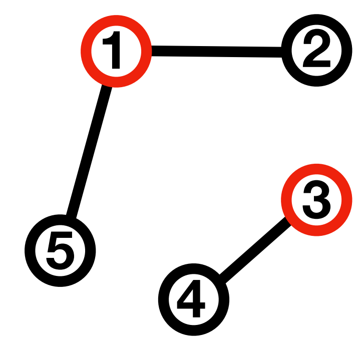
<u>Example:</u> dynamic MIS

Input: Updates to G		Delete (1, 3)	Delete (1, 4)	
Picture				
Output: Changes to the MIS		3 joins MIS	4 joins MIS	

<u>Example:</u> dynamic MIS

Input: Updates to G		Delete (1, 3)	Delete (1, 4)	Insert (3, 4)
Picture				
Output: Changes to the MIS		3 joins MIS	4 joins MIS	

<u>Example:</u> dynamic MIS

Input: Updates to G		Delete (1, 3)	Delete (1, 4)	Insert (3, 4)
Picture				
Output: Changes to the MIS		3 joins MIS	4 joins MIS	4 leaves MIS

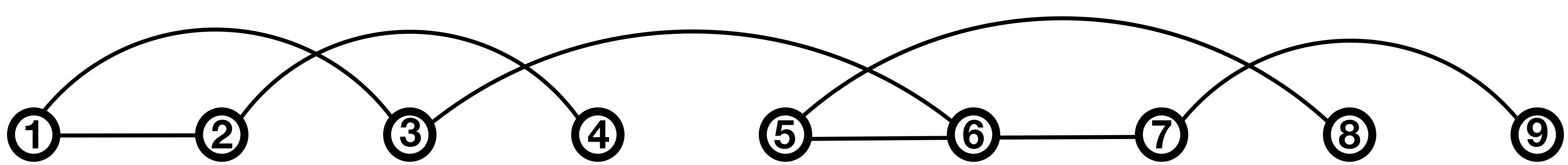
A short history

- n = # of vertices, m = # of edges, Δ = max-deg of G

Ref	[AOSS'18]	[GK'18] [DZ'18]	[DZ'18]	[AOSS'19]	Ours	[BDH+'19]
Update time	$O(m^{3/4})$	$O(m^{2/3})$	$O(m^{1/2})$	$\tilde{O}(n^{1/2})$	$O(\log^4 n)$	$O(\log^2 n \log^2 \Delta)$
Det?	Yes	Yes	No	No	No	No

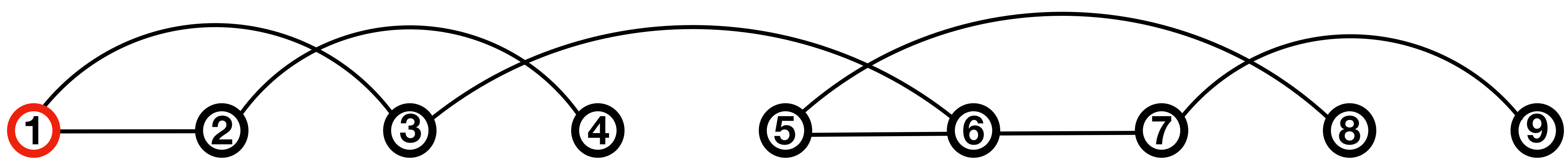
<u>Definition:</u> random-greedy MIS

- Given a random order π of all vertices
- Go over each vertex in order π, add it to MIS if not yet dominated



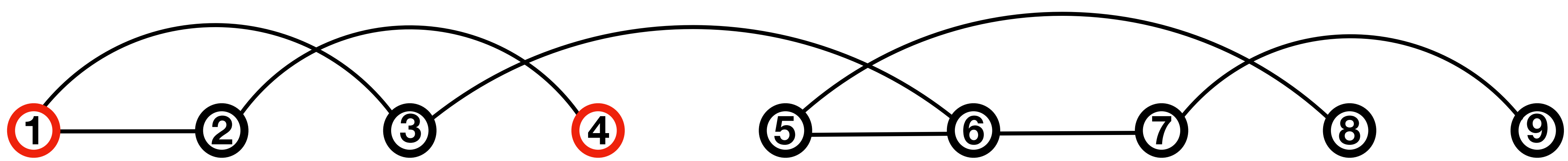
Definition: random-greedy MIS

- Given a random order π of all vertices
- Go over each vertex in order π, add it to MIS if not yet dominated



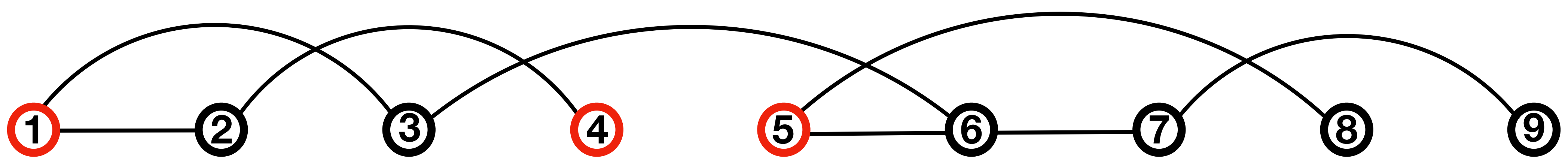
<u>Definition:</u> random-greedy MIS

- Given a random order π of all vertices
- Go over each vertex in order π, add it to MIS if not yet dominated



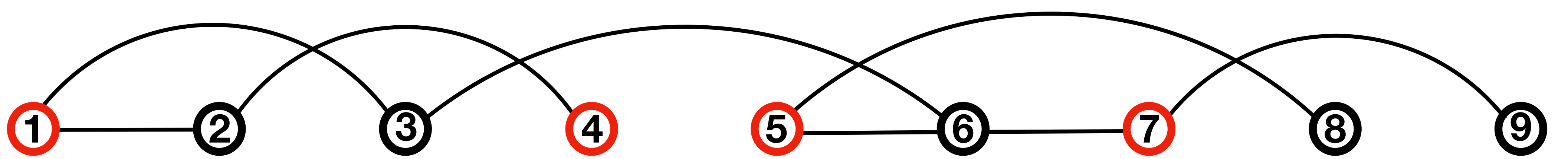
Definition: random-greedy MIS

- Given a random order π of all vertices
- Go over each vertex in order π, add it to MIS if not yet dominated



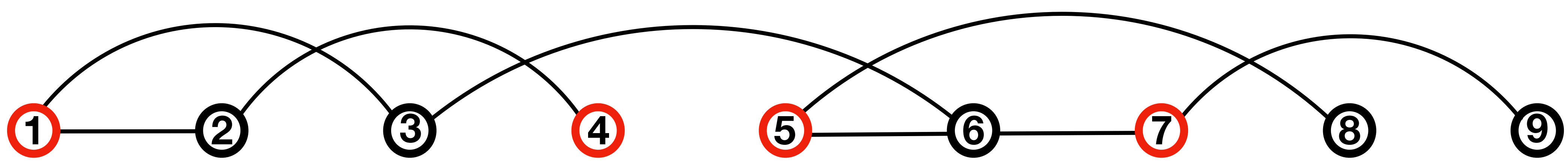
Definition: random-greedy MIS

- Given a random order π of all vertices
- Go over each vertex in order π, add it to MIS if not yet dominated



Definition: random-greedy MIS

- Given a random order π of all vertices
- Go over each vertex in order π, add it to MIS if not yet dominated



Main Theorem:

There is an algorithm that maintains a greedy MIS with respect to π in expected time $O(\log^4 n)$, expectation taken over the uniformly random choice of π

A degree-reduction lemma [AOSS'19]

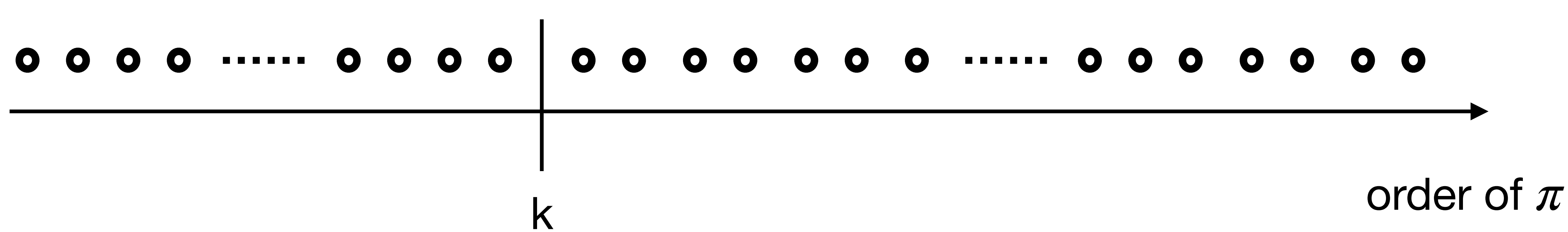
Lemma:

Let $U \subseteq V$ be the set of vertices not dominated by vertices whose order $\leq k$, then $\Delta(G[U]) \leq \tilde{O}(n/k)$ w.h.p. over π

A degree-reduction lemma [AOSS'19]

<u>Lemma:</u>

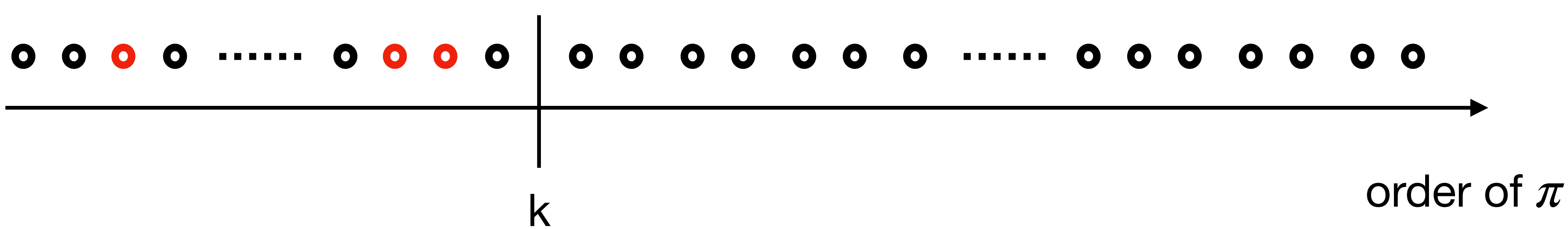
Let $U \subseteq V$ be the set of vertices not dominated by vertices whose order $\leq k$, then $\Delta(G[U]) \leq \tilde{O}(n/k)$ w.h.p. over π



A degree-reduction lemma [AOSS'19]

Lemma:

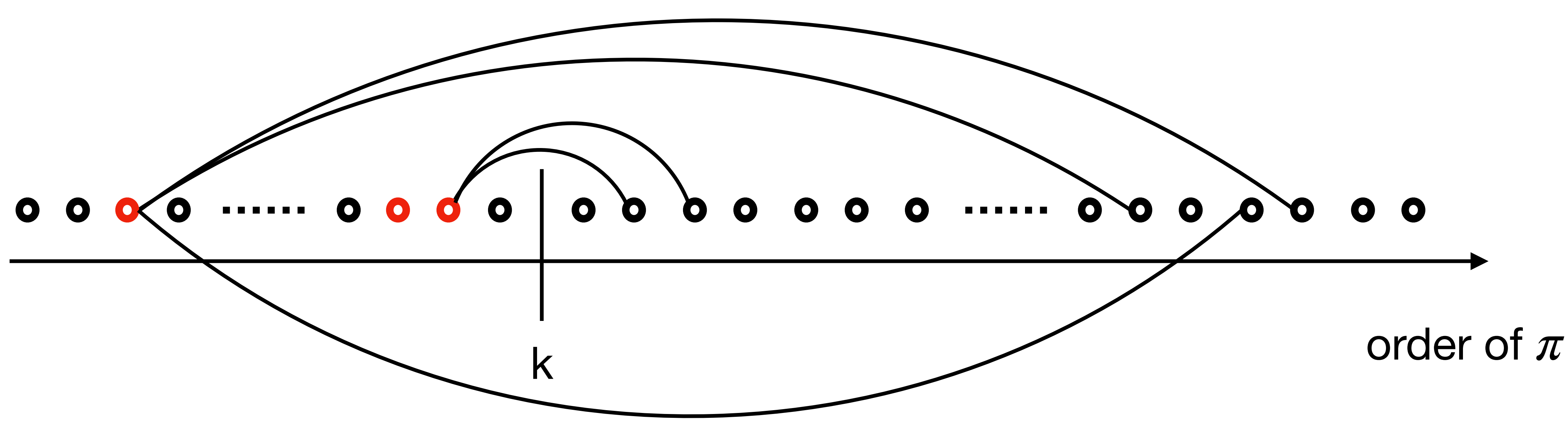
Let $U \subseteq V$ be the set of vertices not dominated by vertices whose order $\leq k$, then $\Delta(G[U]) \leq \tilde{O}(n/k)$ w.h.p. over π



A degree-reduction lemma [AOSS'19]

Lemma:

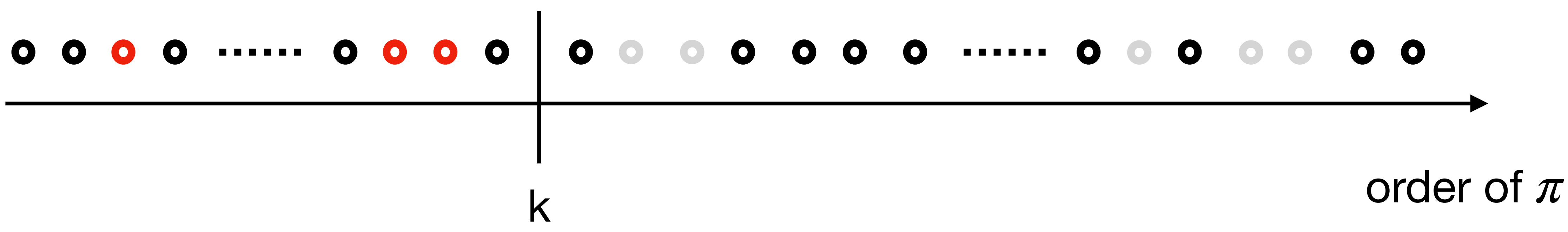
Let $U \subseteq V$ be the set of vertices not dominated by vertices whose order $\leq k$, then $\Delta(G[U]) \leq \tilde{O}(n/k)$ w.h.p. over π



A degree-reduction lemma [AOSS'19]

Lemma:

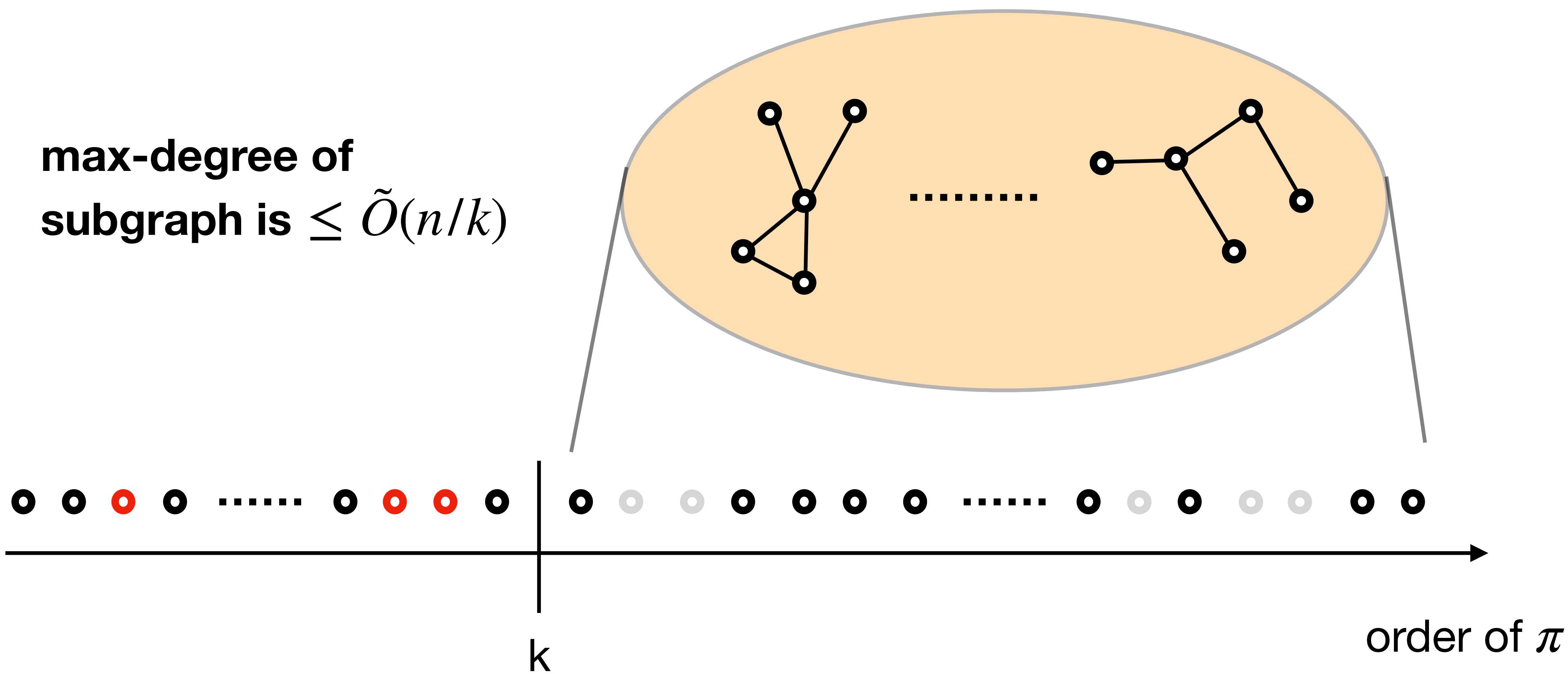
Let $U \subseteq V$ be the set of vertices not dominated by vertices whose order $\leq k$, then $\Delta(G[U]) \leq \tilde{O}(n/k)$ w.h.p. over π



A degree-reduction lemma [AOSS'19]

Lemma:

Let $U \subseteq V$ be the set of vertices not dominated by vertices whose order $\leq k$, then $\Delta(G[U]) \leq \tilde{O}(n/k)$ w.h.p. over π



A structural lemma [CHK'16]

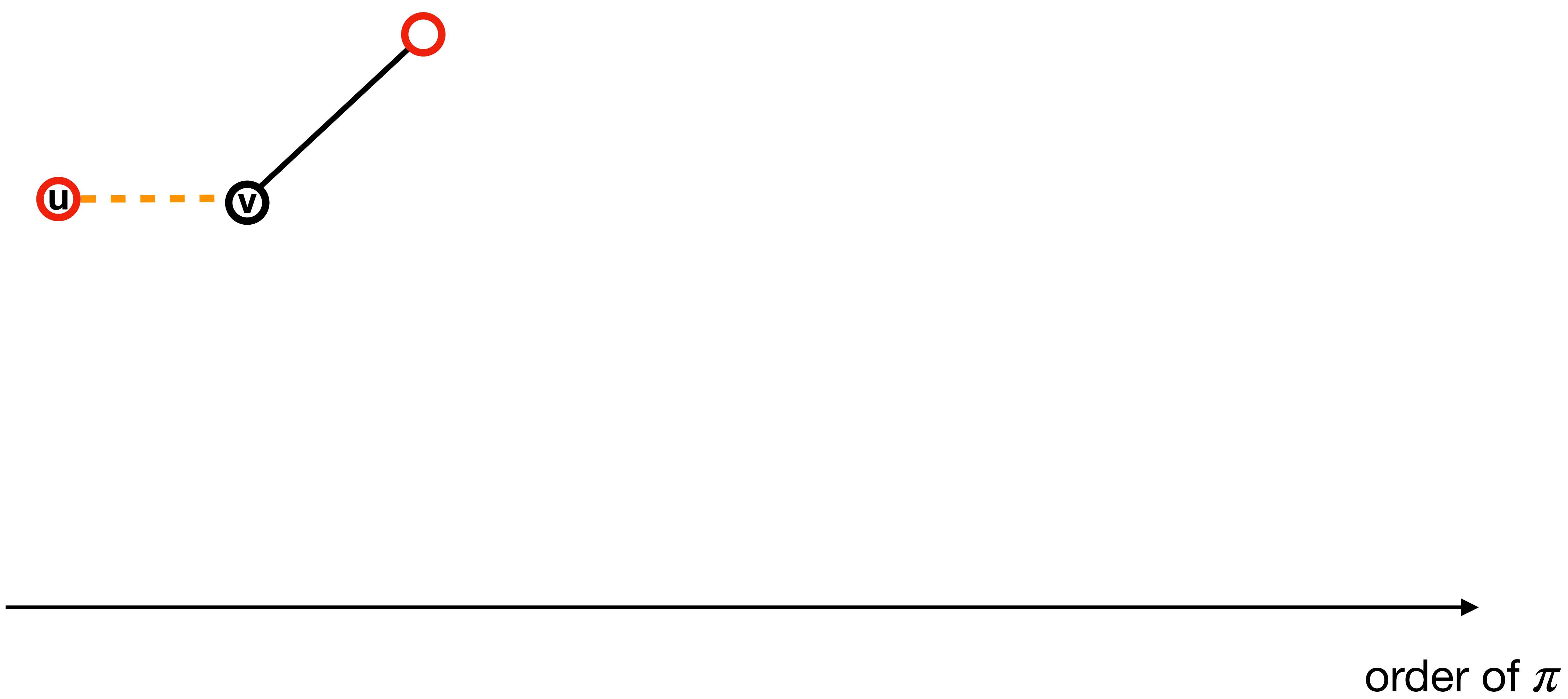
Definition:

Suppose an edge **(u, v)** is updated. Define the following **influenced set S**

A structural lemma [CHK'16]

Definition:

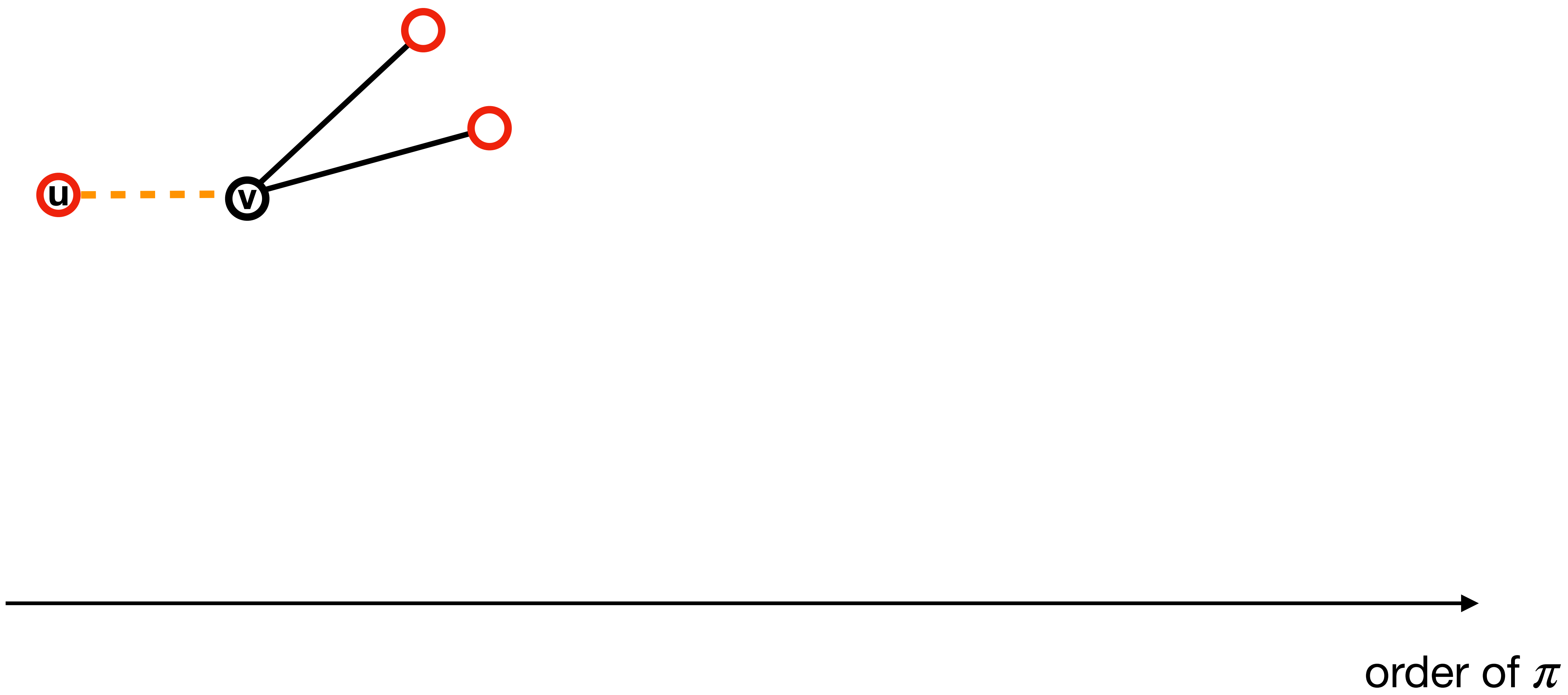
Suppose an edge **(u, v)** is updated. Define the following **influenced set S**



A structural lemma [CHK'16]

<u>Definition:</u>

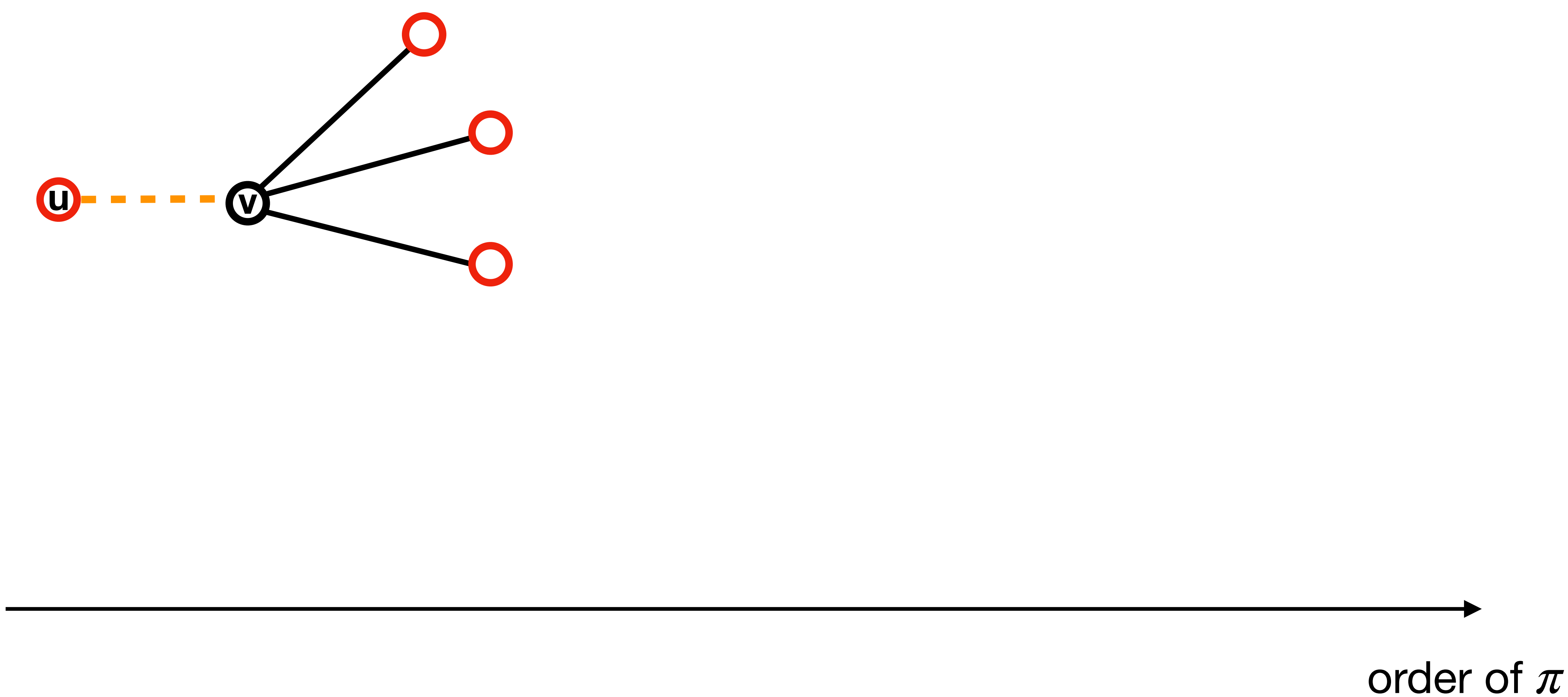
Suppose an edge **(u, v)** is updated. Define the following **influenced set S**



A structural lemma [CHK'16]

Definition:

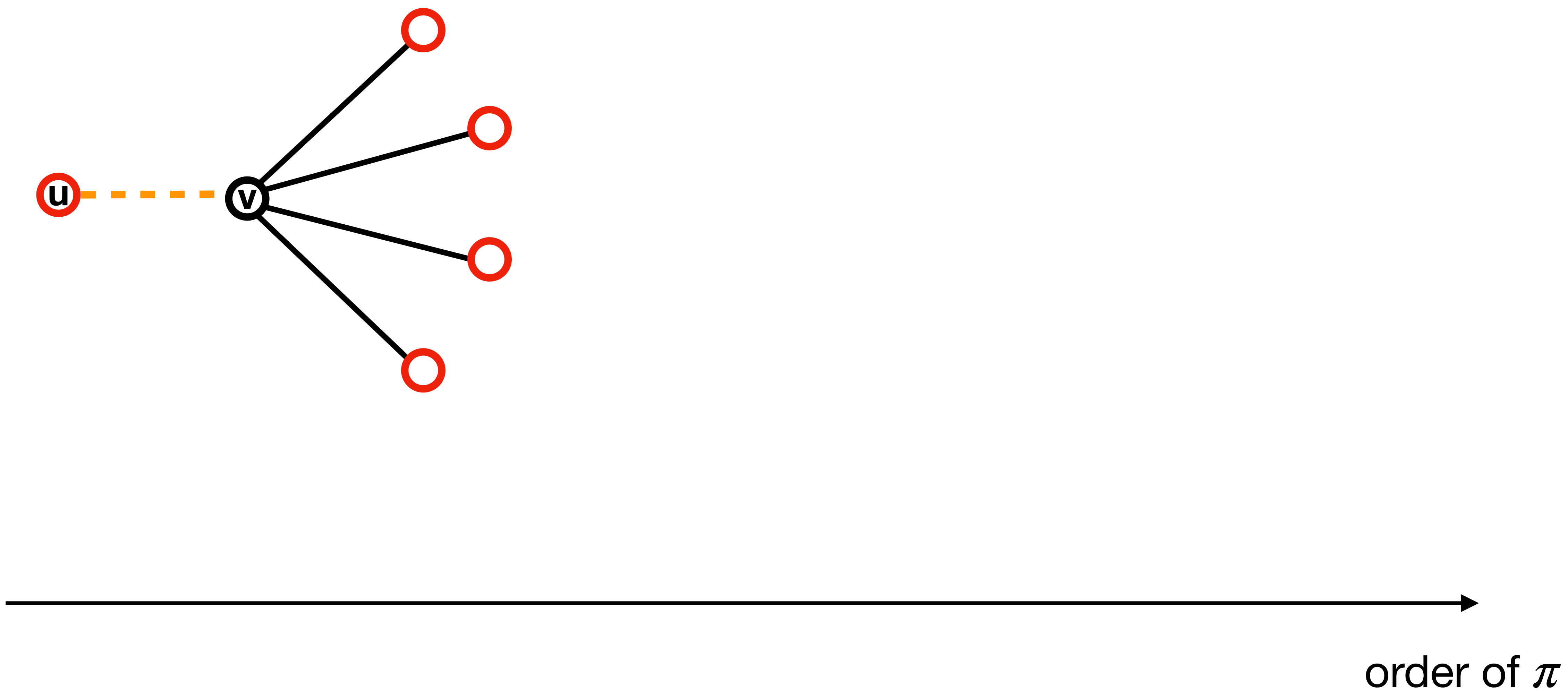
Suppose an edge **(u, v)** is updated. Define the following **influenced set S**



A structural lemma [CHK'16]

Definition:

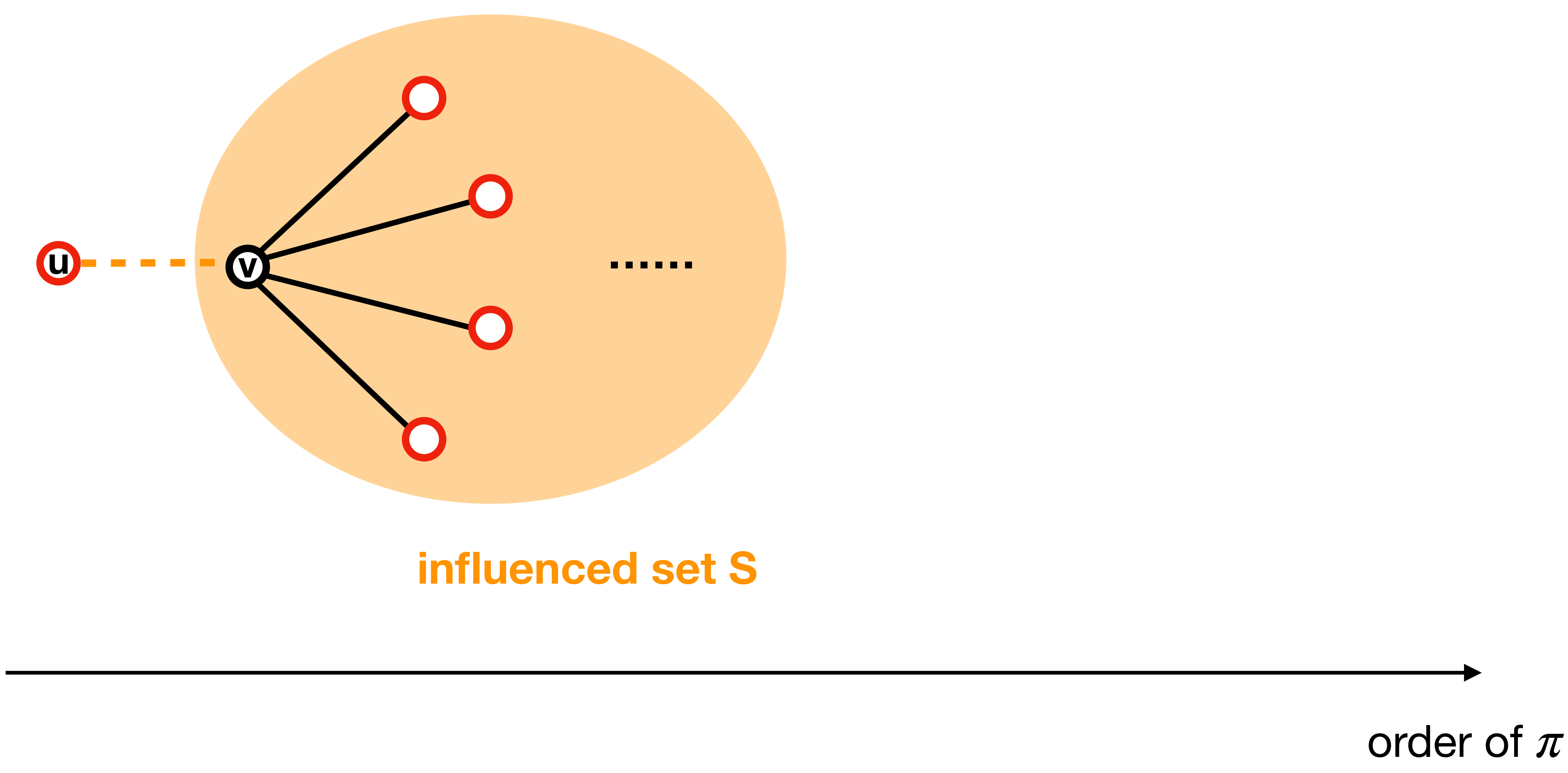
Suppose an edge **(u, v)** is updated. Define the following **influenced set S**



A structural lemma [CHK'16]

<u>Definition:</u>

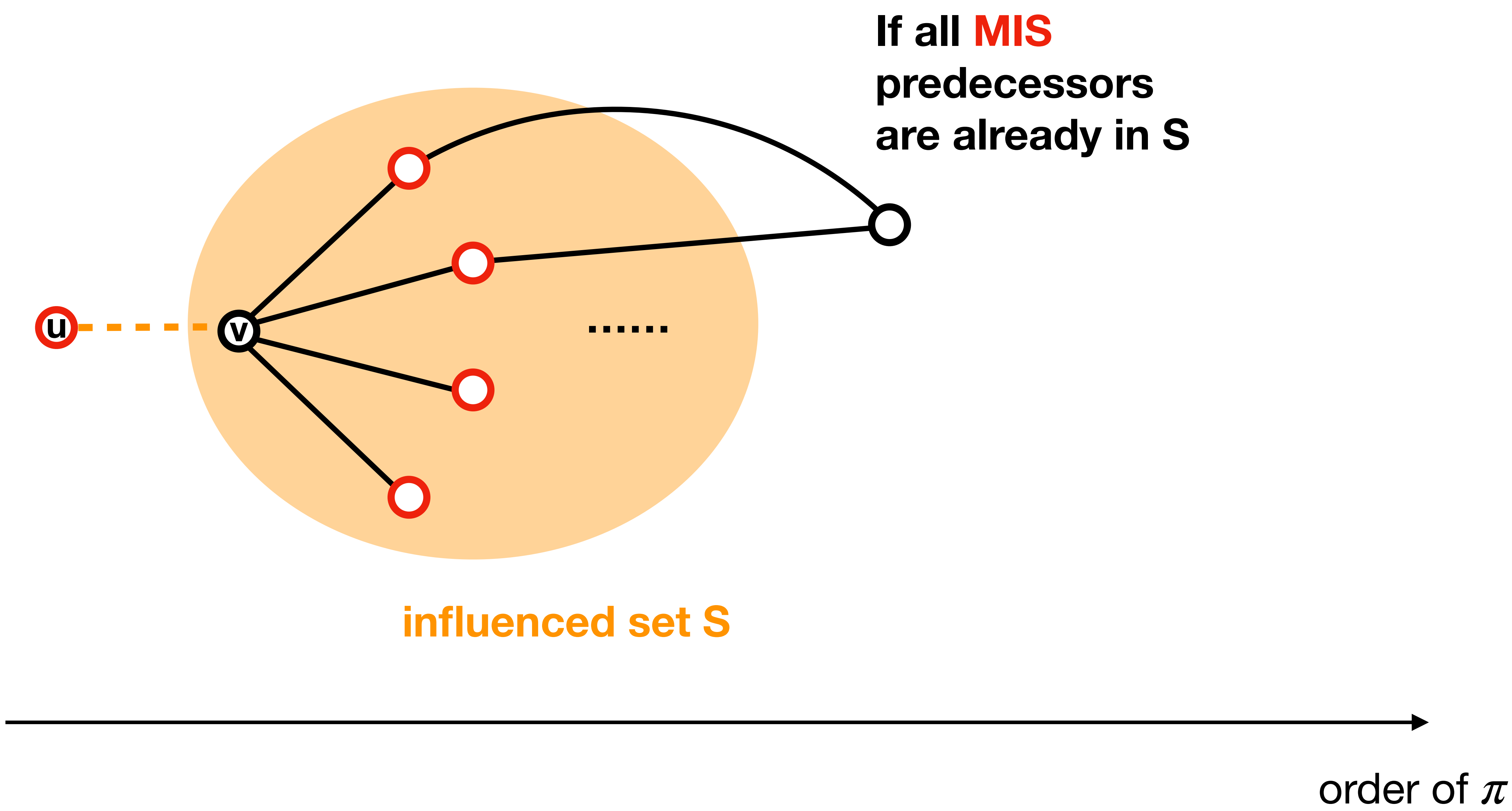
Suppose an edge **(u, v)** is updated. Define the following **influenced set S**



A structural lemma [CHK'16]

<u>Definition:</u>

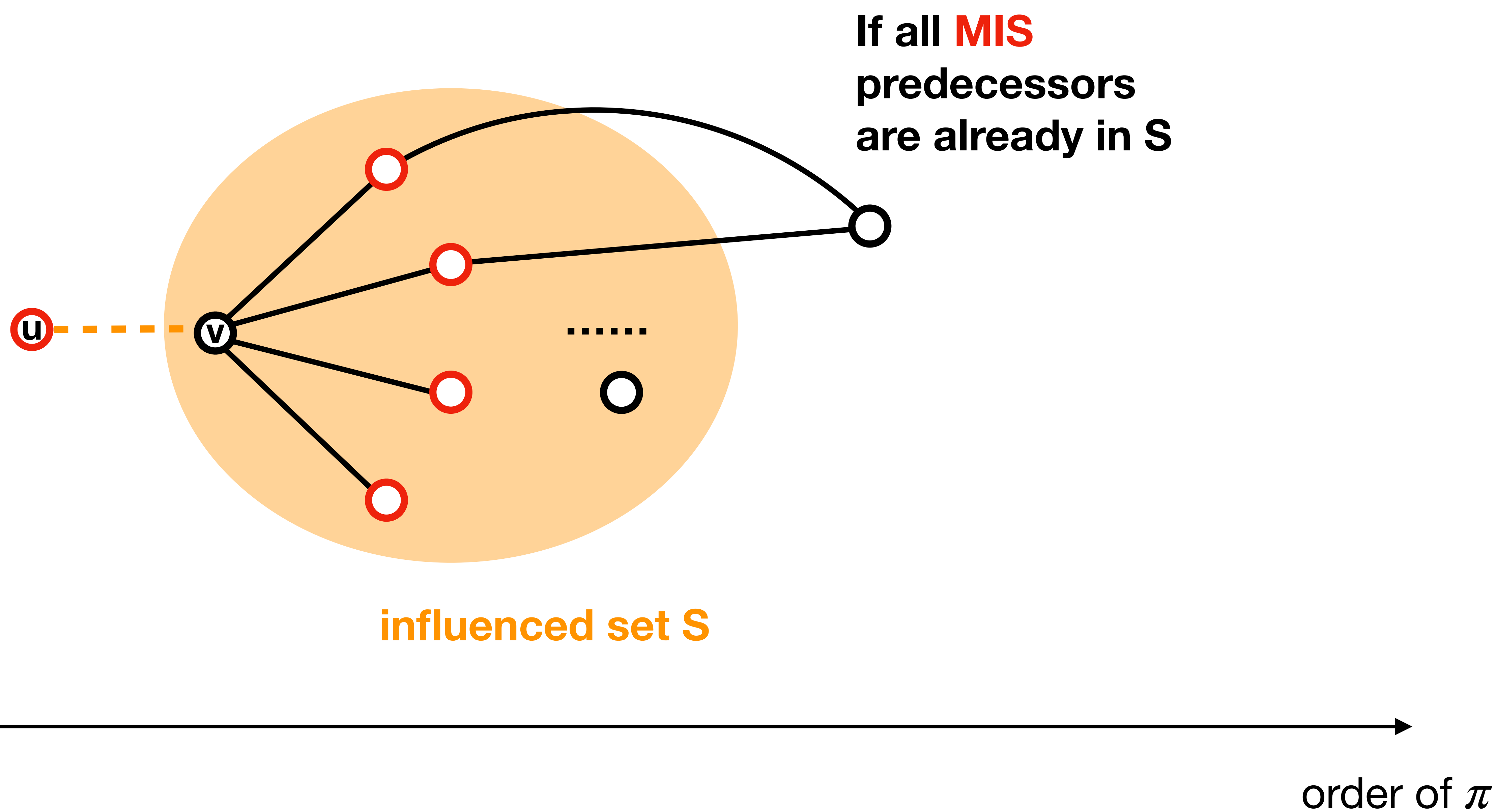
Suppose an edge **(u, v)** is updated. Define the following **influenced set S**



A structural lemma [CHK'16]

Definition:

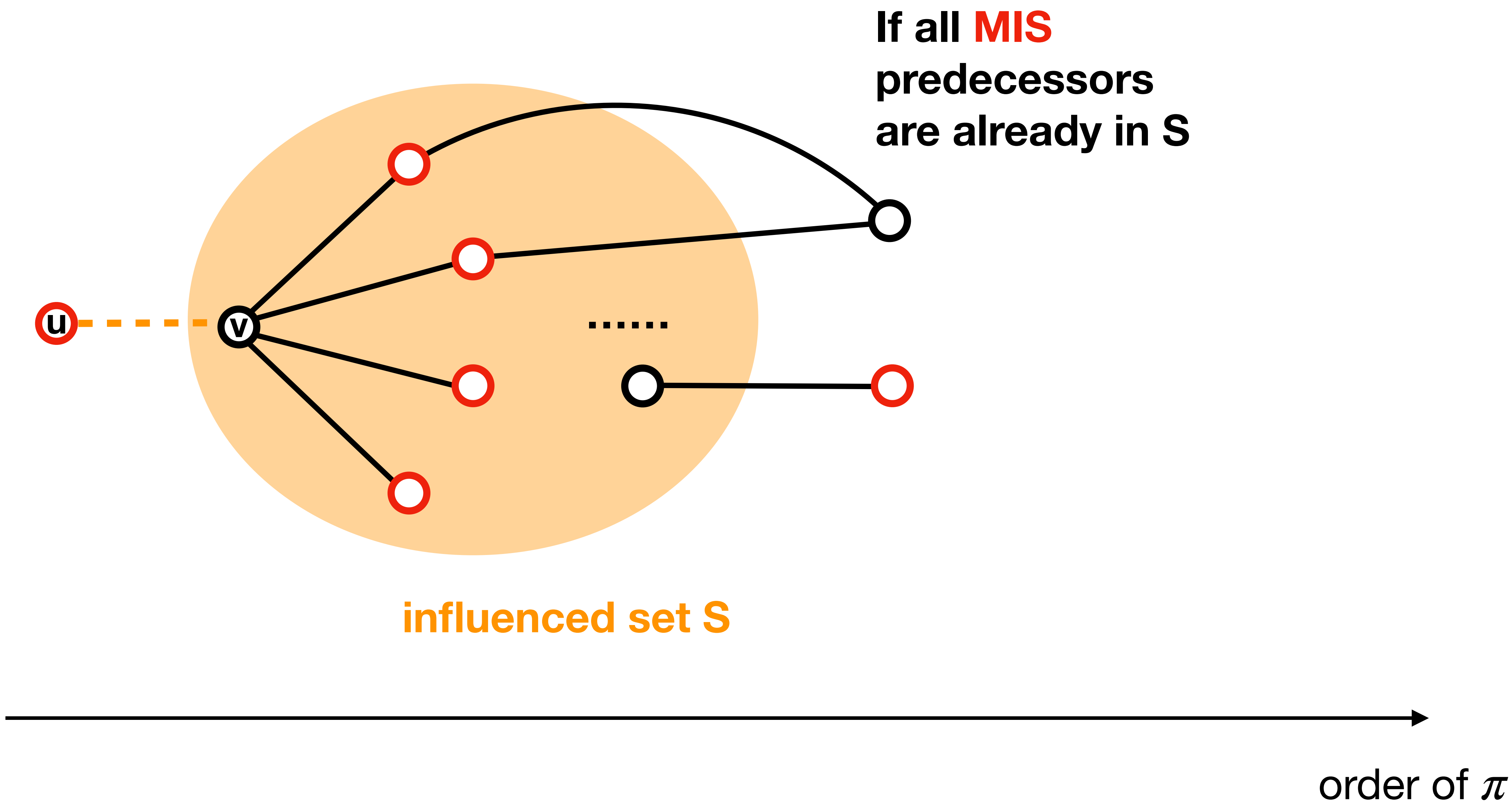
Suppose an edge **(u, v)** is updated. Define the following **influenced set S**



A structural lemma [CHK'16]

Definition:

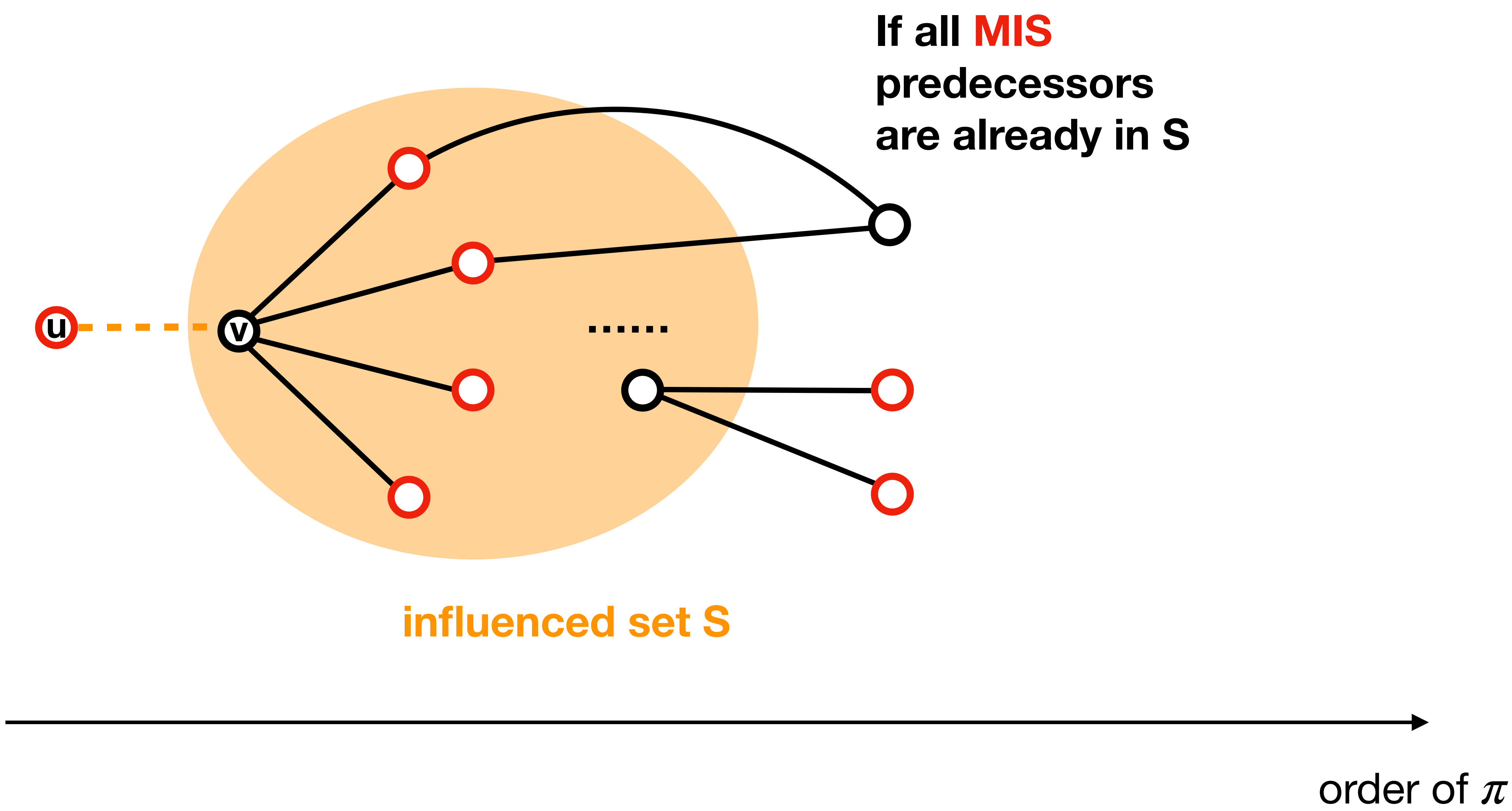
Suppose an edge **(u, v)** is updated. Define the following **influenced set S**



A structural lemma [CHK'16]

Definition:

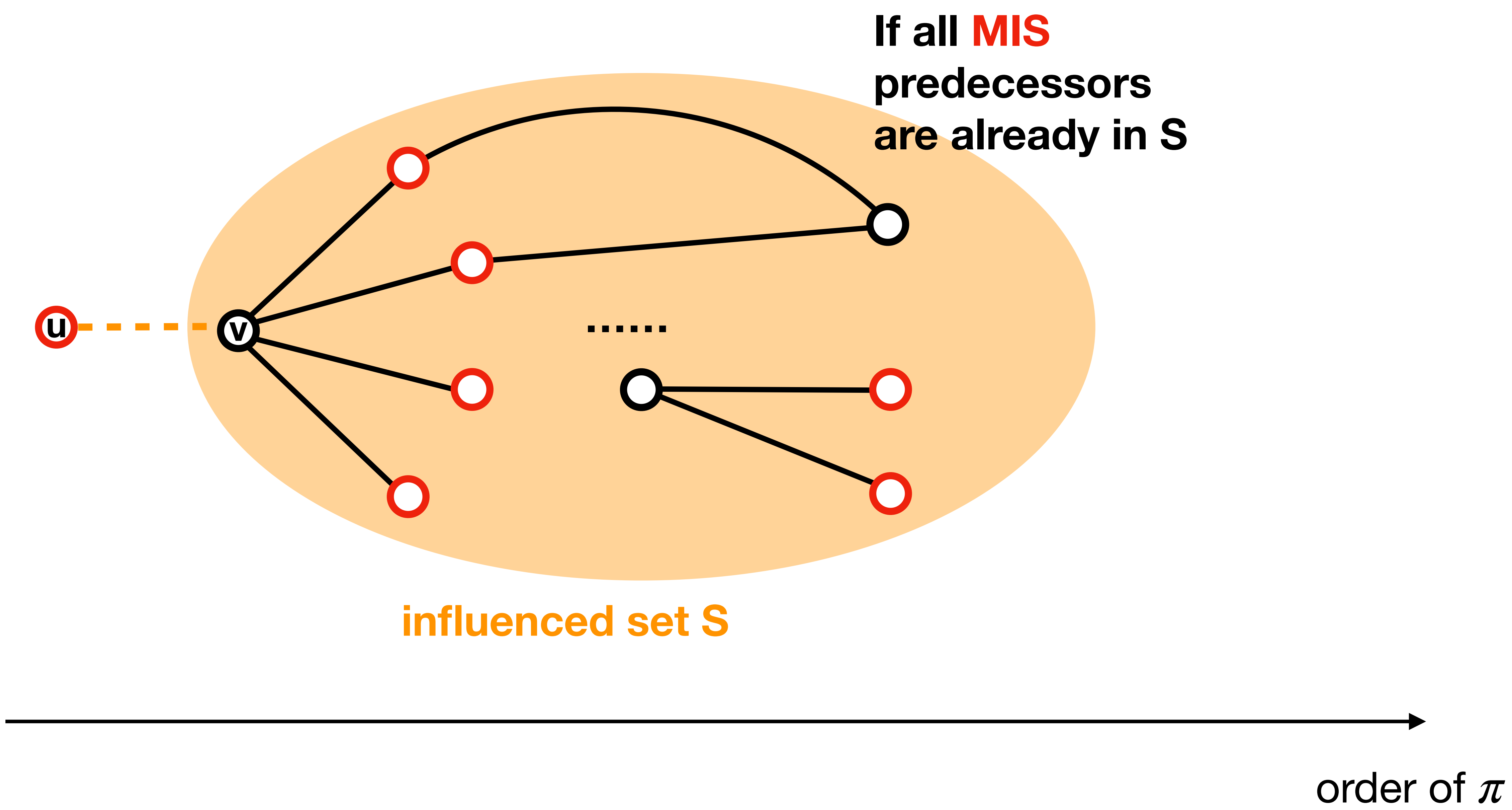
Suppose an edge **(u, v)** is updated. Define the following **influenced set S**



A structural lemma [CHK'16]

Definition:

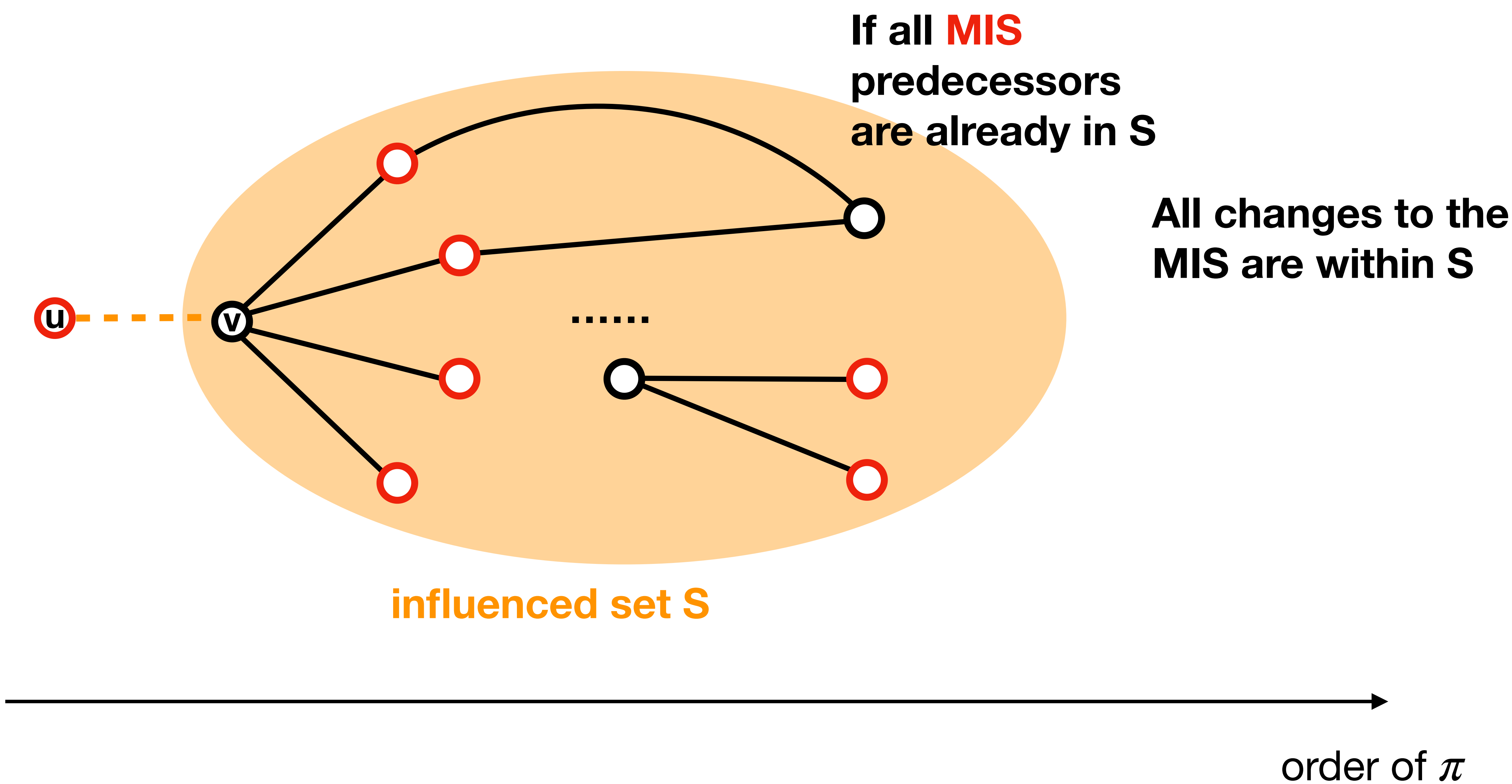
Suppose an edge **(u, v)** is updated. Define the following **influenced set S**



A structural lemma [CHK'16]

<u>Definition:</u>

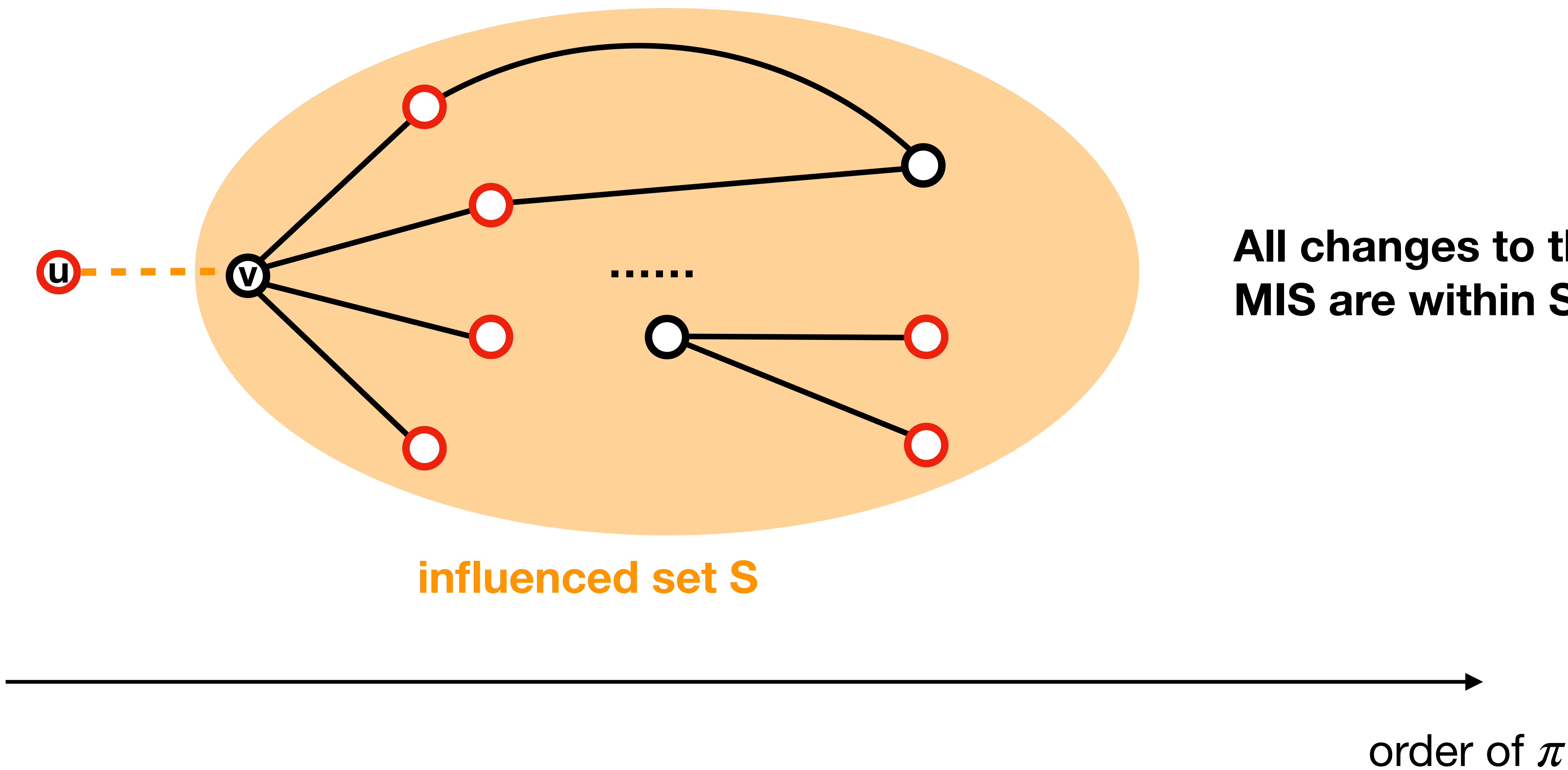
Suppose an edge **(u, v)** is updated. Define the following **influenced set S**



A structural lemma [CHK'16]

<u>Lemma [CHK'16]:</u>

Suppose an edge **(u, v)** is updated. Then $\mathrm{E}_{\pi}[\,|S|\,] \leq O(1)$, over uniformly random choices of π

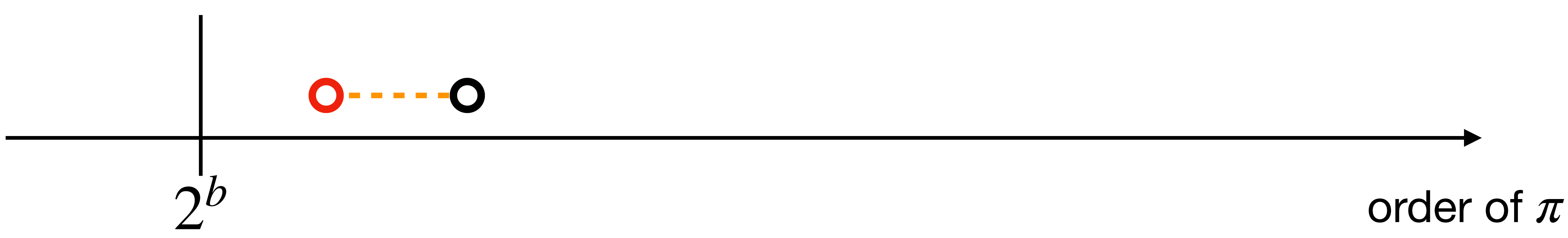


All changes to the MIS are within S

A sketch of algorithm

Data structure:

For each 2^b, explicitly maintain the induced subgraph G_b of vertices not dominated by MIS vertices of order $\leq 2^b$



A sketch of algorithm

<u>Data structure:</u>

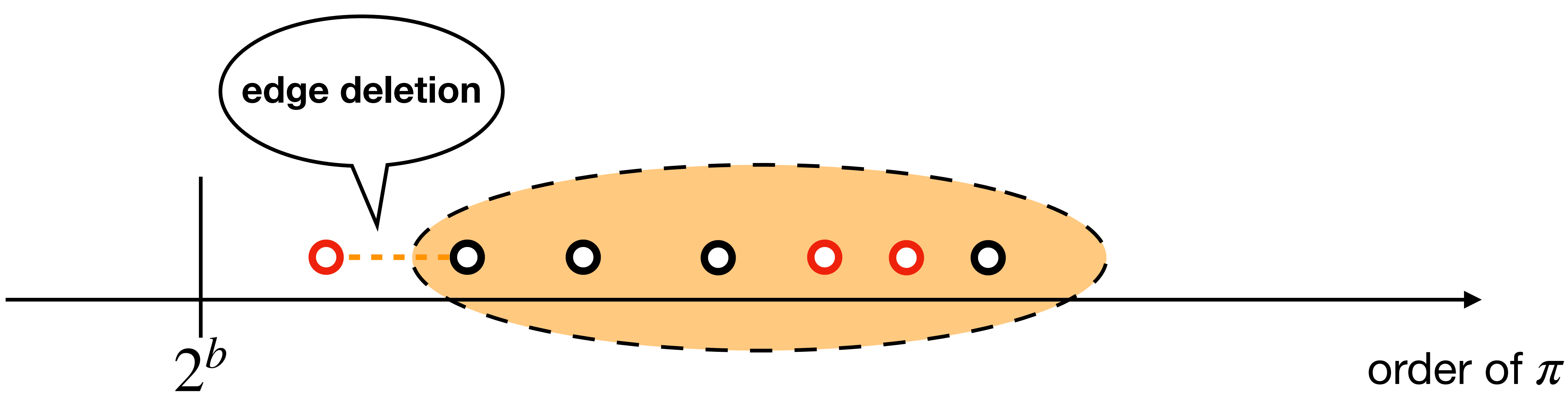
For each 2^b, explicitly maintain the induced subgraph G_b of vertices not dominated by MIS vertices of order $\leq 2^b$

A sketch of algorithm

Data structure:

For each 2^b, explicitly maintain the induced subgraph G_b of vertices not dominated by MIS vertices of order $\leq 2^b$

(1) Compute the **influenced set**



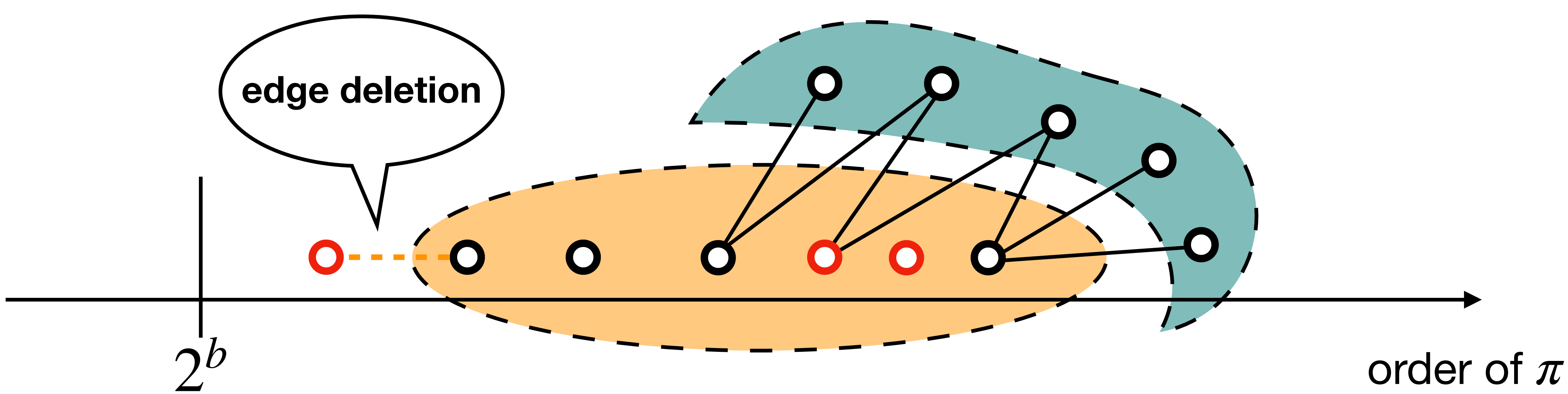
A sketch of algorithm

<u>Data structure:</u>

For each 2^b, explicitly maintain the induced subgraph G_b of vertices not dominated by MIS vertices of order $\leq 2^b$

(1) Compute the **influenced set**

(2) Compute the **vertex updates to subgraphs**



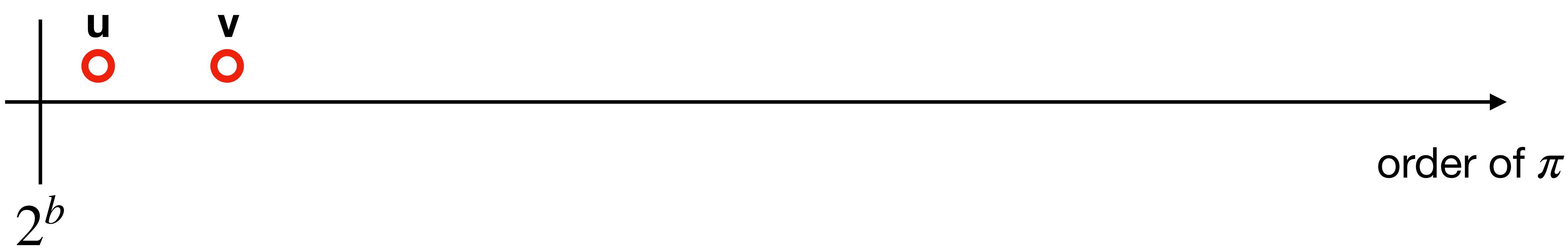
The main algorithm

Data structure:

For each 2^b, explicitly maintain the induced subgraph G_b of vertices not dominated by MIS vertices of order $\leq 2^b$

Update algorithm:

(1) Compute the **influenced set S**



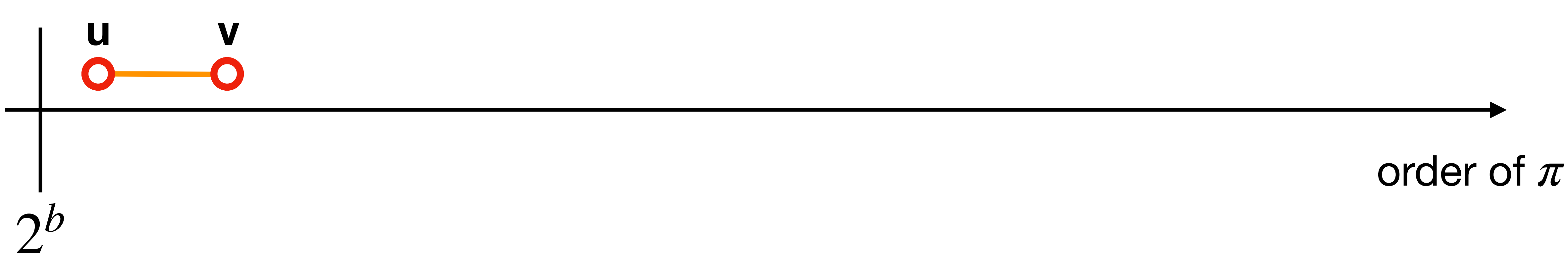
The main algorithm

Data structure:

For each 2^b, explicitly maintain the induced subgraph G_b of vertices not dominated by MIS vertices of order $\leq 2^b$

Update algorithm:

(1) Compute the **influenced set S**



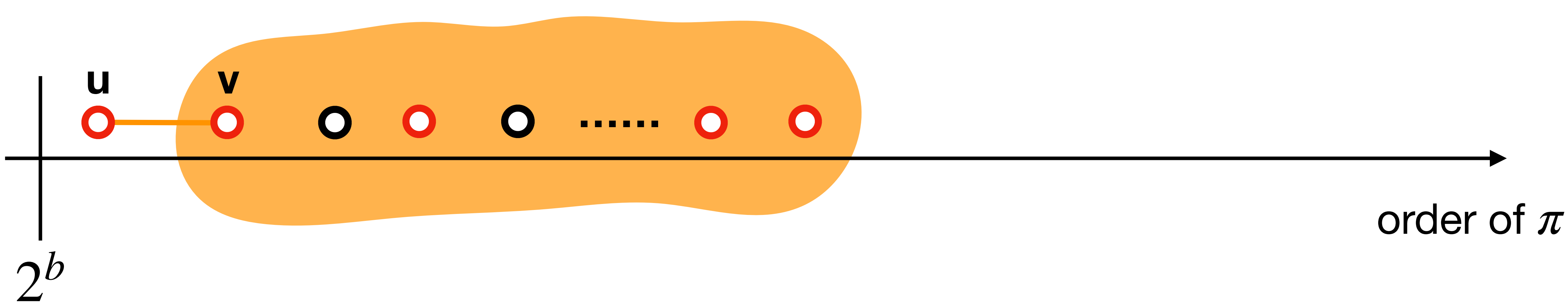
The main algorithm

<u>Data structure:</u>

For each 2^b, explicitly maintain the induced subgraph G_b of vertices not dominated by MIS vertices of order $\leq 2^b$

<u>Update algorithm:</u>

(1) Compute the **influenced set S**



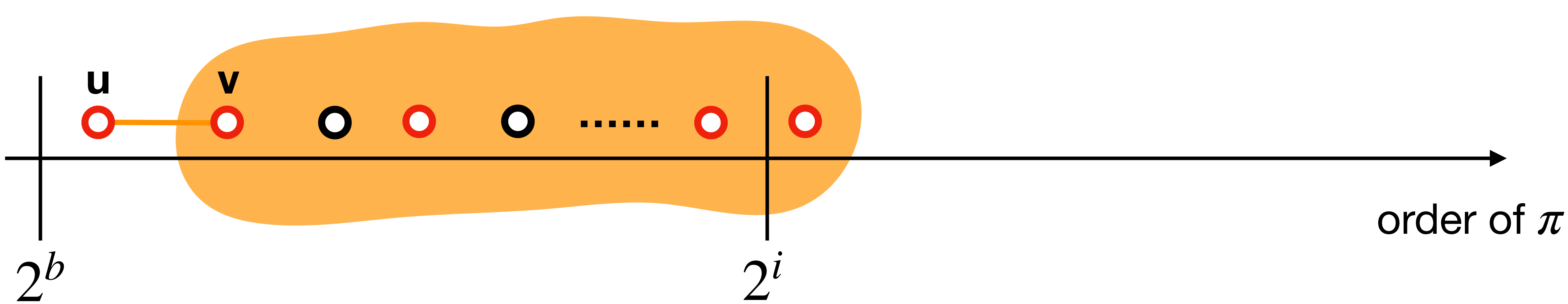
The main algorithm

Data structure:

For each 2^b, explicitly maintain the induced subgraph G_b of vertices not dominated by MIS vertices of order $\leq 2^b$

Update algorithm:

(1) Compute the **influenced set S**



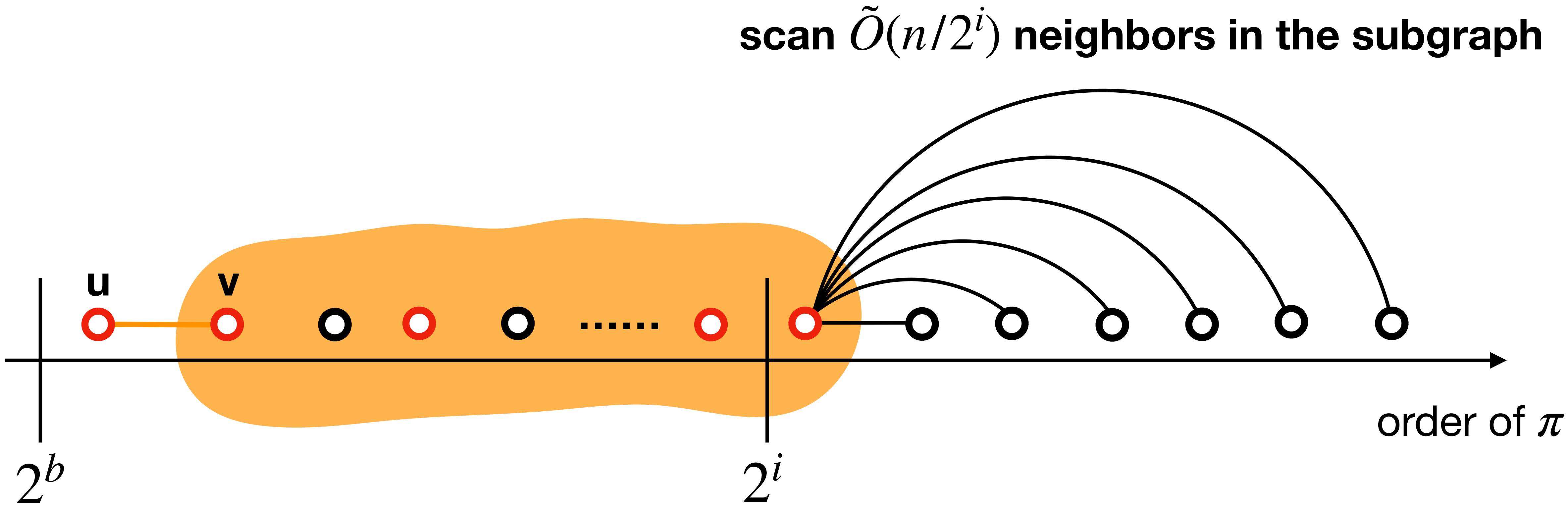
The main algorithm

Data structure:

For each 2^b, explicitly maintain the induced subgraph G_b of vertices not dominated by MIS vertices of order $\leq 2^b$

Update algorithm:

(1) Compute the **influenced set S**



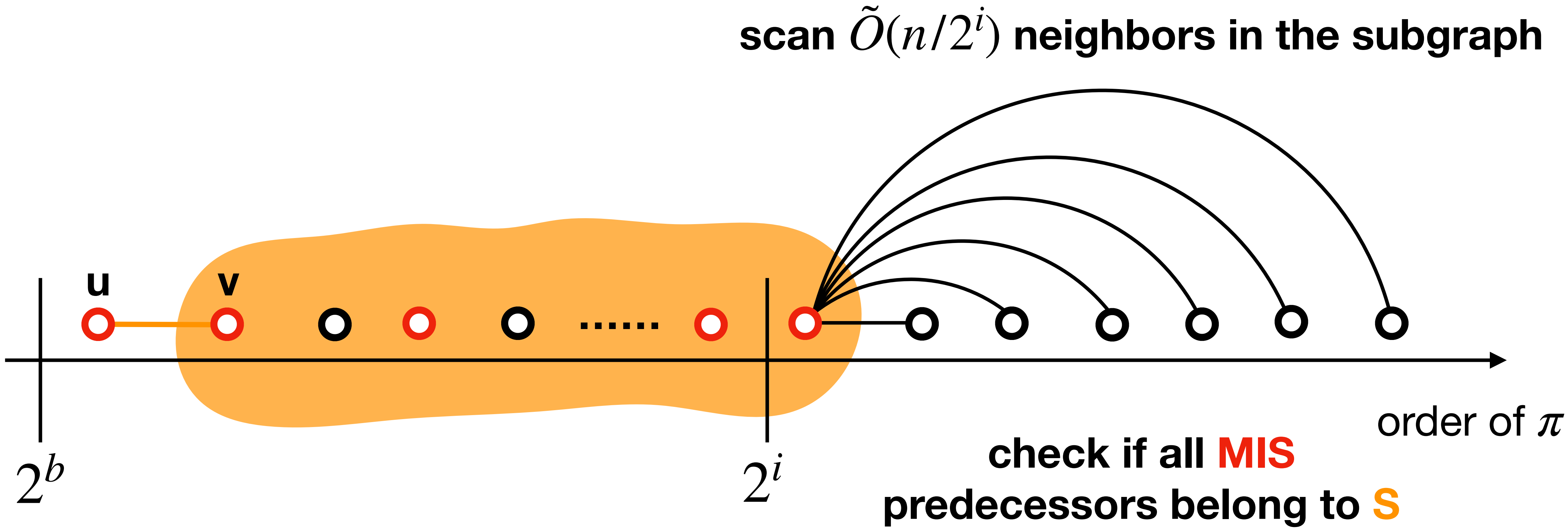
The main algorithm

Data structure:

For each 2^b, explicitly maintain the induced subgraph G_b of vertices not dominated by MIS vertices of order $\leq 2^b$

Update algorithm:

(1) Compute the **influenced set S**



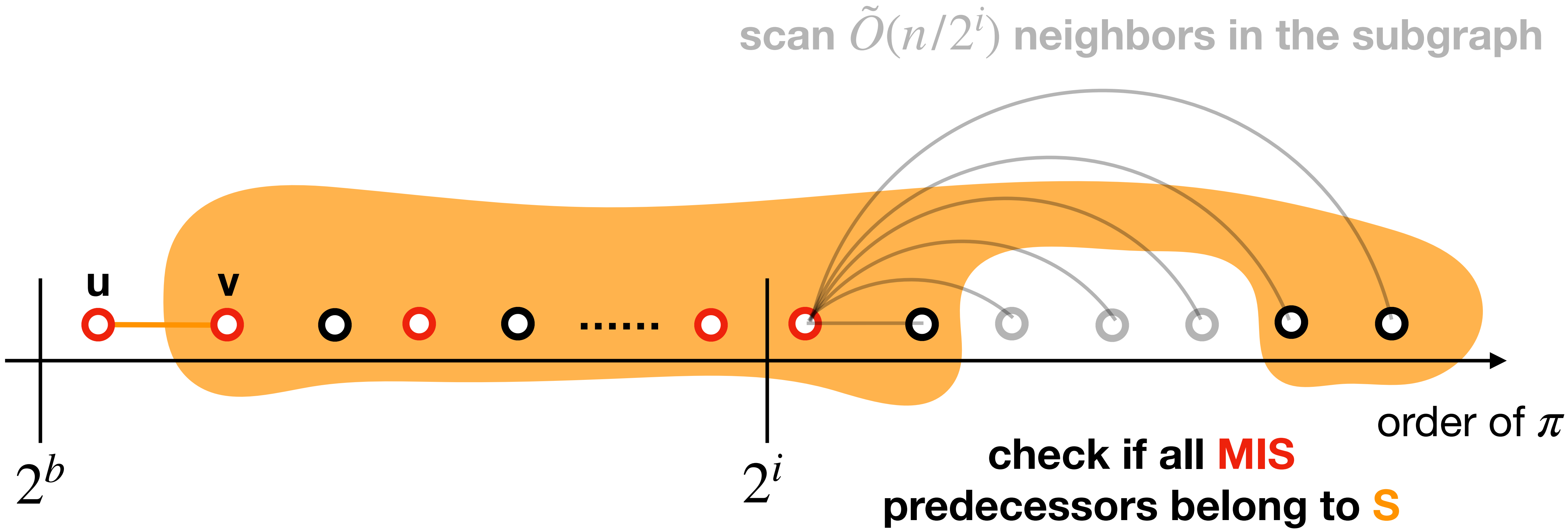
The main algorithm

Data structure:

For each 2^b, explicitly maintain the induced subgraph G_b of vertices not dominated by MIS vertices of order $\leq 2^b$

Update algorithm:

(1) Compute the **influenced set S**



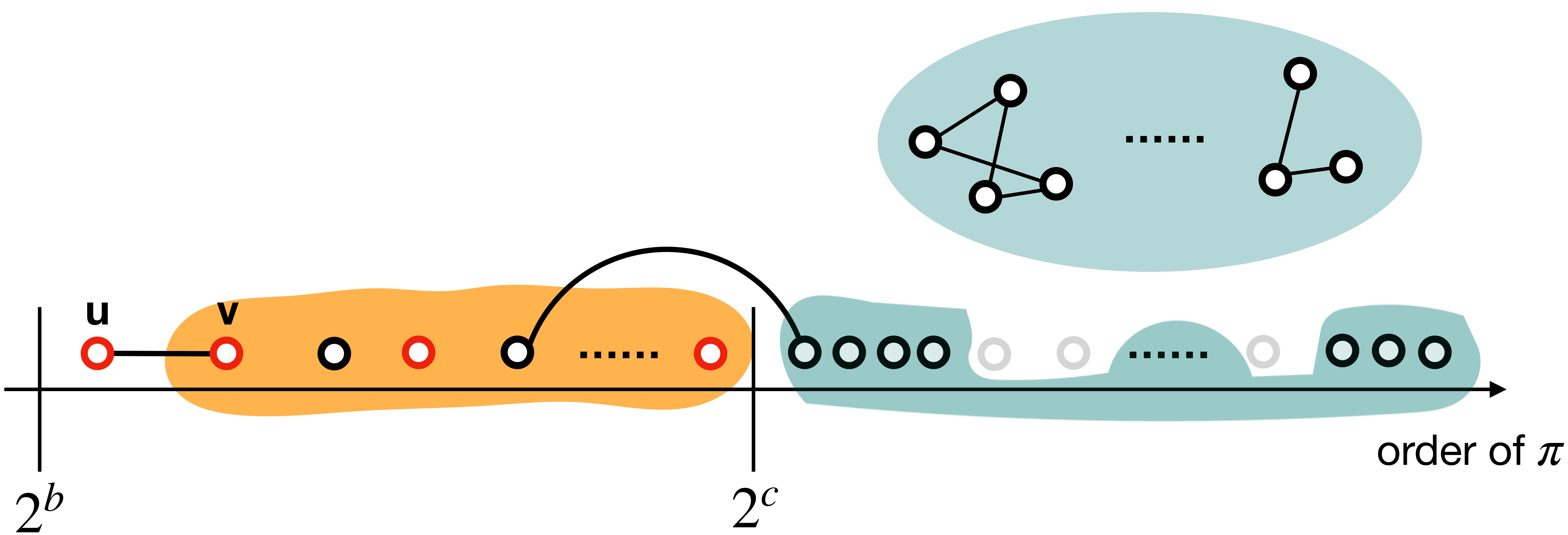
The main algorithm

Data structure:

For each 2^b, explicitly maintain the induced subgraph G_b of vertices not dominated by MIS vertices of order $\leq 2^b$

Update algorithm:

(2) Update all induced subgraphs $G_{c(>b)}$



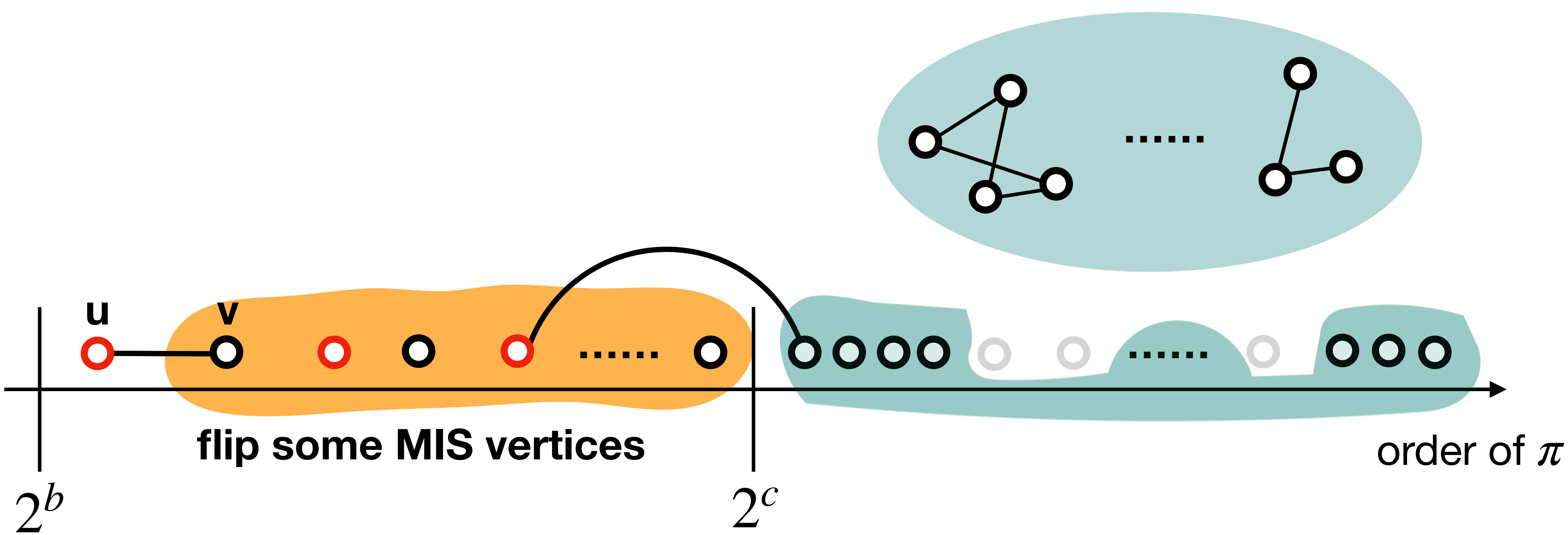
The main algorithm

<u>Data structure:</u>

For each 2^b, explicitly maintain the induced subgraph G_b of vertices not dominated by MIS vertices of order $\leq 2^b$

<u>Update algorithm:</u>

(2) Update all induced subgraphs $G_{c(>b)}$



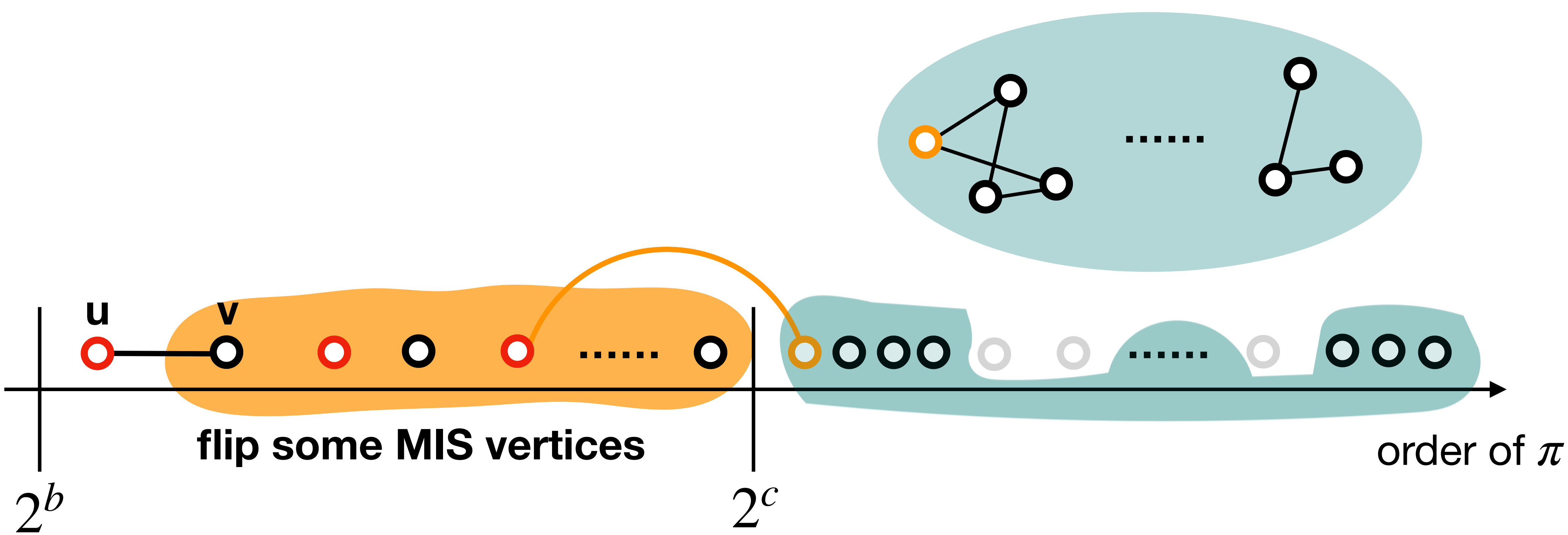
The main algorithm

Data structure:

For each 2^b, explicitly maintain the induced subgraph G_b of vertices not dominated by MIS vertices of order $\leq 2^b$

Update algorithm:

(2) Update all induced subgraphs $G_{c(>b)}$



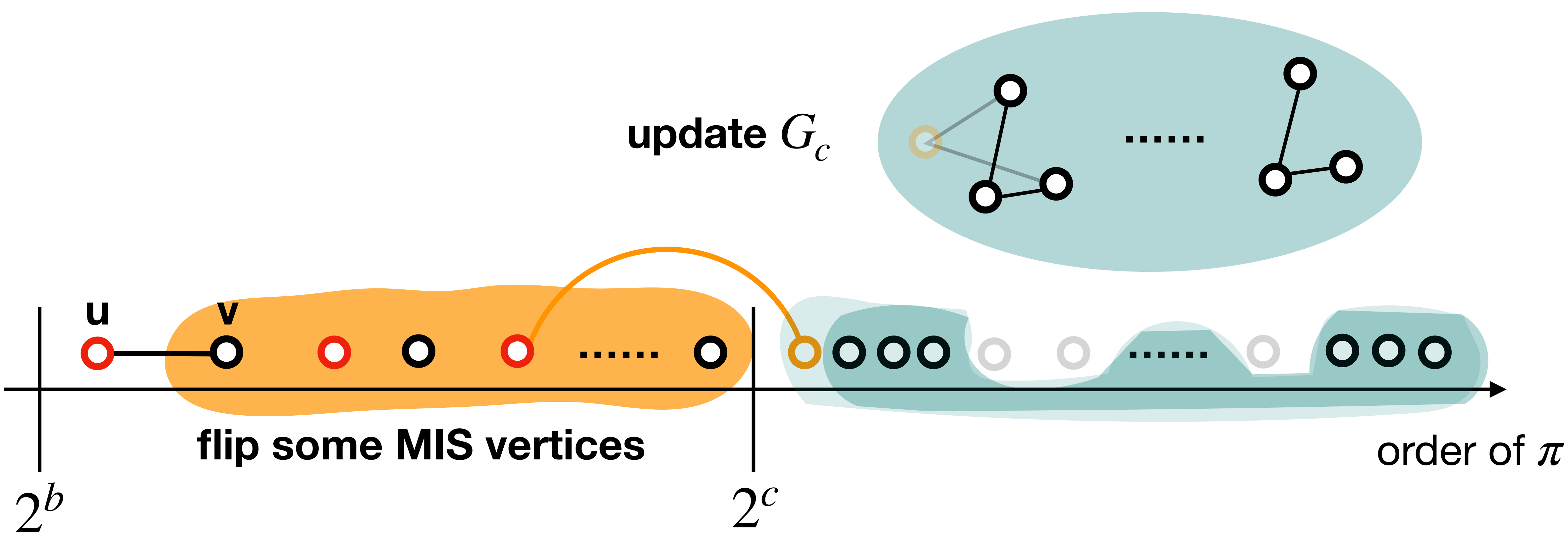
The main algorithm

Data structure:

For each 2^b, explicitly maintain the induced subgraph G_b of vertices not dominated by MIS vertices of order $\leq 2^b$

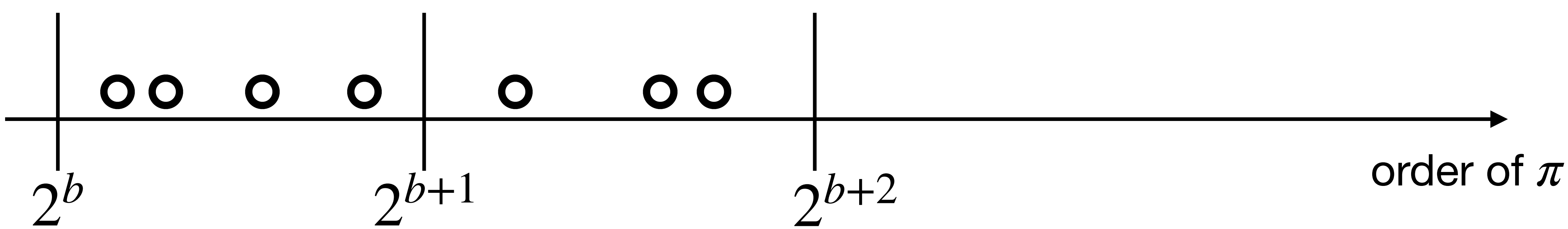
Update algorithm:

(2) Update all induced subgraphs $G_{c(>b)}$



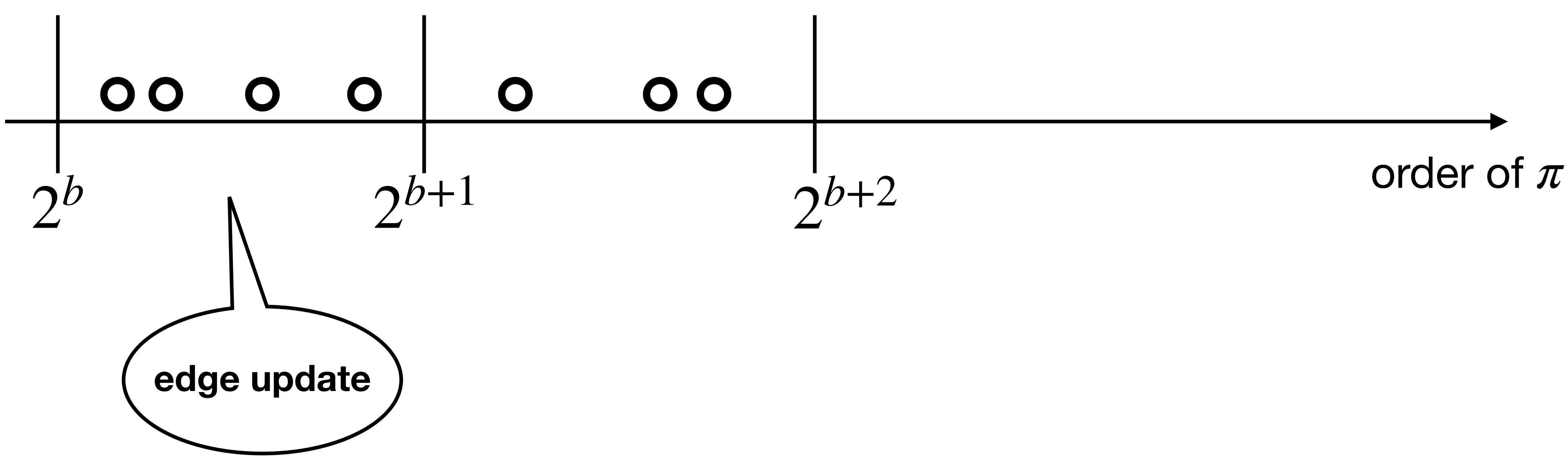
A sketch of analysis

Say $2^b < \pi(u) < \pi(v) \leq 2^{b+1}$, want update time $\leq \tilde{O}(n^2/2^{2b})$



A sketch of analysis

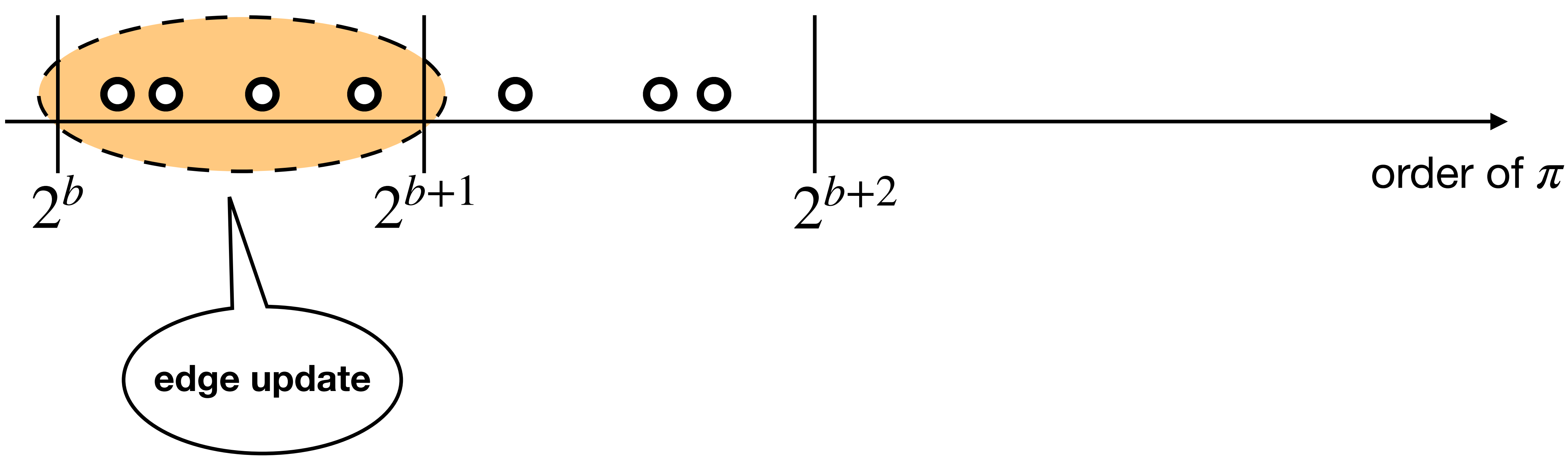
Say $2^b < \pi(u) < \pi(v) \leq 2^{b+1}$, want update time $\leq \tilde{O}(n^2/2^{2b})$



A sketch of analysis

Say $2^b < \pi(u) < \pi(v) \leq 2^{b+1}$, want update time $\leq \tilde{O}(n^2/2^{2b})$

 # MIS-flips [CHK'16]
$\leq \mathrm{E}_\pi[\,|S|\,] \leq 1$



A sketch of analysis

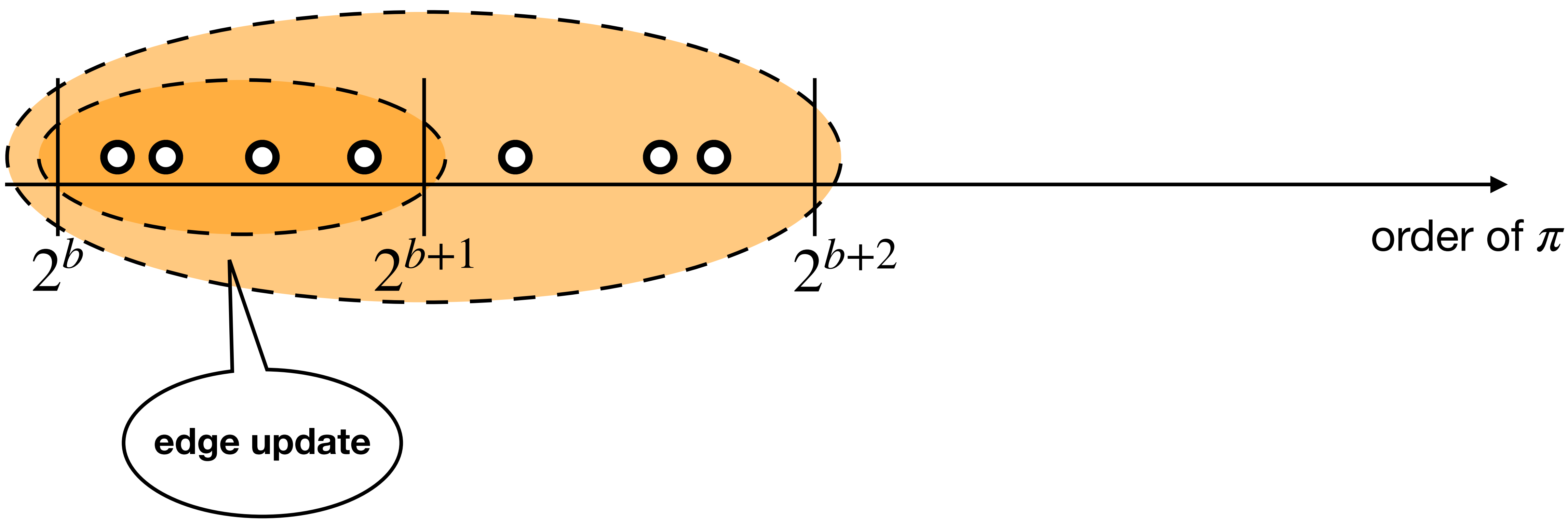
Say $2^b < \pi(u) < \pi(v) \leq 2^{b+1}$, want update time $\leq \tilde{O}(n^2/2^{2b})$

 # MIS-flips [CHK'16]

$\leq \mathrm{E}_\pi[\,|S|\mid$ **first half**$] \leq 2 \cdot \mathrm{E}_\pi[\,|S|\,] \leq 2$

 # MIS-flips [CHK'16]

$\leq \mathrm{E}_\pi[\,|S|\,] \leq 1$



A sketch of analysis

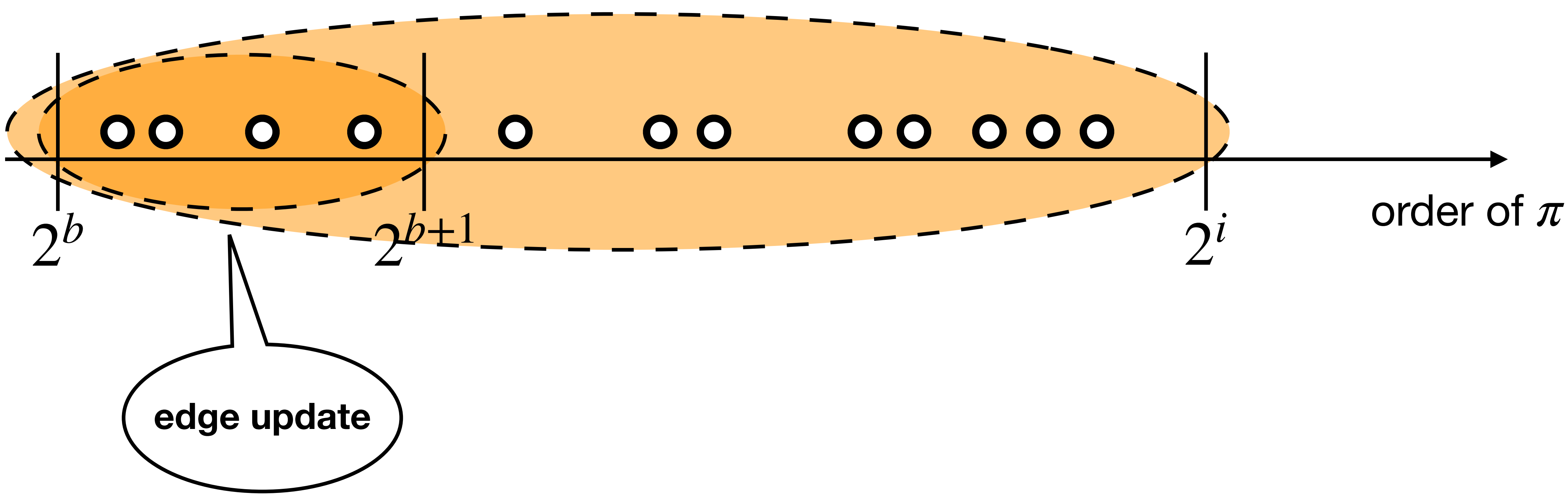
Say $2^b < \pi(u) < \pi(v) \leq 2^{b+1}$, want update time $\leq \tilde{O}(n^2/2^{2b})$

 # MIS-flips [CHK'16]

$\leq \mathrm{E}_\pi[\,|S|\,|$ **first** $2^b] \leq 2^{i-b} \cdot \mathrm{E}_\pi[\,|S|\,] \leq 2^{i-b}$

 # MIS-flips [CHK'16]

$\leq \mathrm{E}_\pi[\,|S|\,] \leq 1$



A sketch of analysis

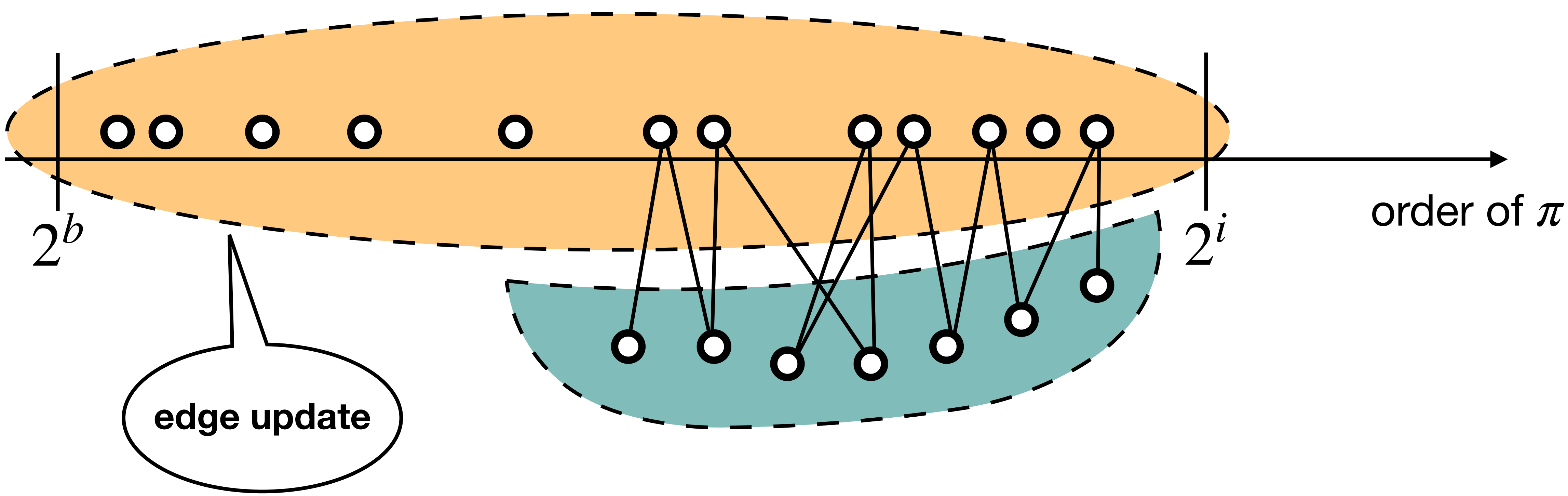
Say $2^b < \pi(u) < \pi(v) \leq 2^{b+1}$, want update time $\leq \tilde{O}(n^2/2^{2b})$

 # MIS-flips [CHK'16]

$\leq \mathrm{E}_\pi[\,|S|\mid$ **first** $2^b] \leq 2^{i-b} \cdot \mathrm{E}_\pi[\,|S|\,] \leq 2^{i-b}$

 # vertex updates to subgraphs [AOSS'19]

$\leq$ **max-deg of** $G_i \cdot \mathrm{E}_\pi[\,|S|\mid$ **first** $2^b] \leq n/2^i \cdot 2^{i-b} = n/2^b$



A sketch of analysis

Say $2^b < \pi(u) < \pi(v) \leq 2^{b+1}$, want update time $\leq \tilde{O}(n^2/2^{2b})$

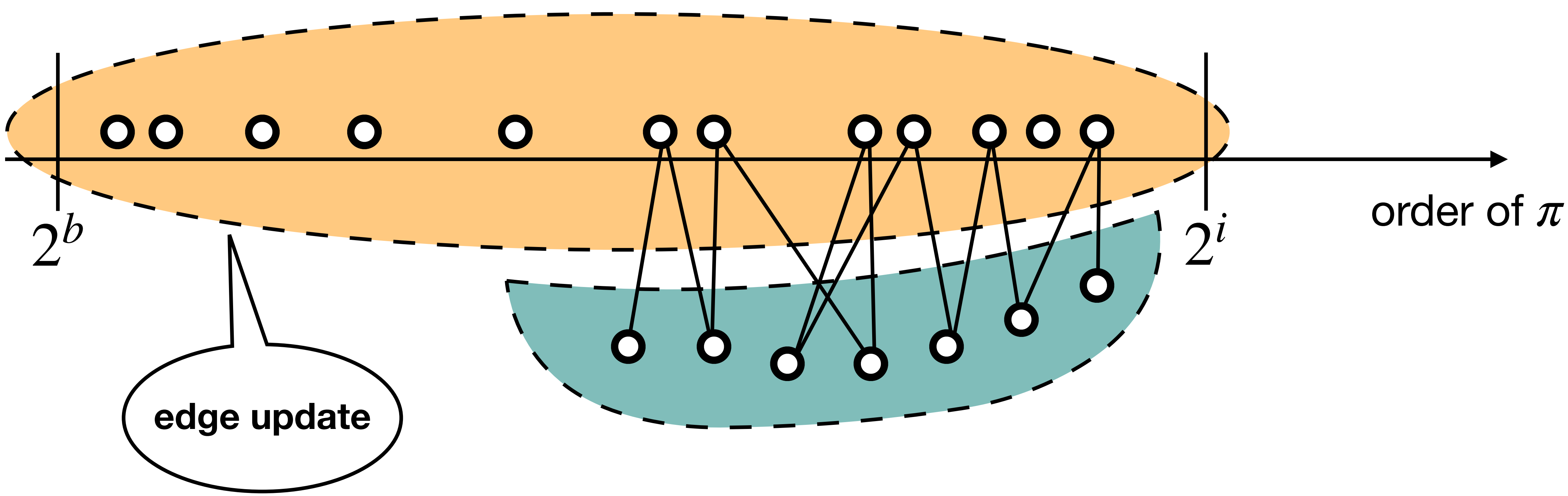
 # MIS-flips [CHK'16]

$\leq \mathrm{E}_\pi[\,|S|\mid$ **first** $2^b] \leq 2^{i-b} \cdot \mathrm{E}_\pi[\,|S|\,] \leq 2^{i-b}$

 # vertex updates to subgraphs [AOSS'19]

$\leq$ **max-deg of** $G_i \cdot \mathrm{E}_\pi[\,|S|\mid$ **first** $2^b] \leq n/2^i \cdot 2^{i-b} = n/2^b$

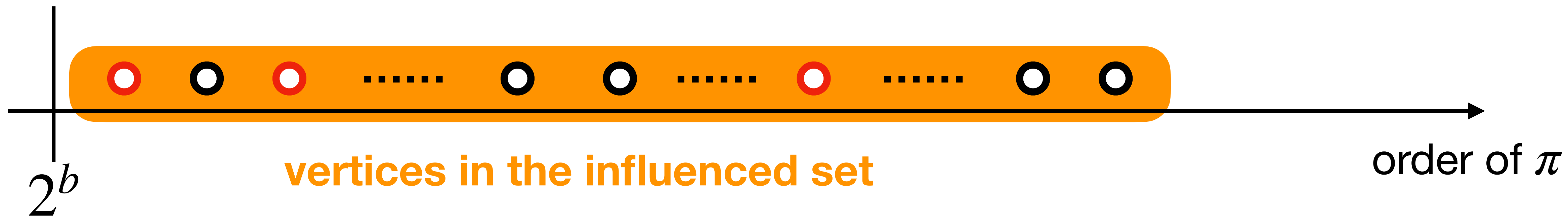
total work $\leq$ (**# subgraph changes**) $\cdot$ **max-deg of** $G_b \leq n^2/2^{2b}$



Running time analysis

Say $2^b < \pi(u) < \pi(v) \leq 2^{b+1}$, want update time $\leq \tilde{O}(n^2/2^{2b})$

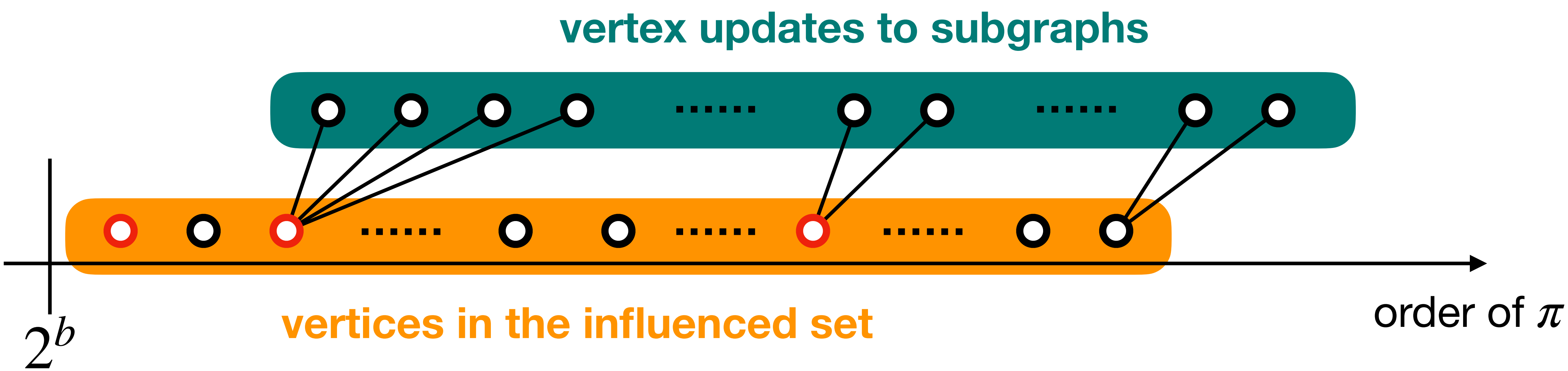
Extending [CHK'16], we could prove that $\mathrm{E}_\pi[\,|S|\,] \leq O(n/2^b)$



Running time analysis

Say $2^b < \pi(u) < \pi(v) \leq 2^{b+1}$, want update time $\leq \tilde{O}(n^2/2^{2b})$

Extending [CHK'16], we could prove that $E_\pi[\,|S|\,] \leq O(n/2^b)$

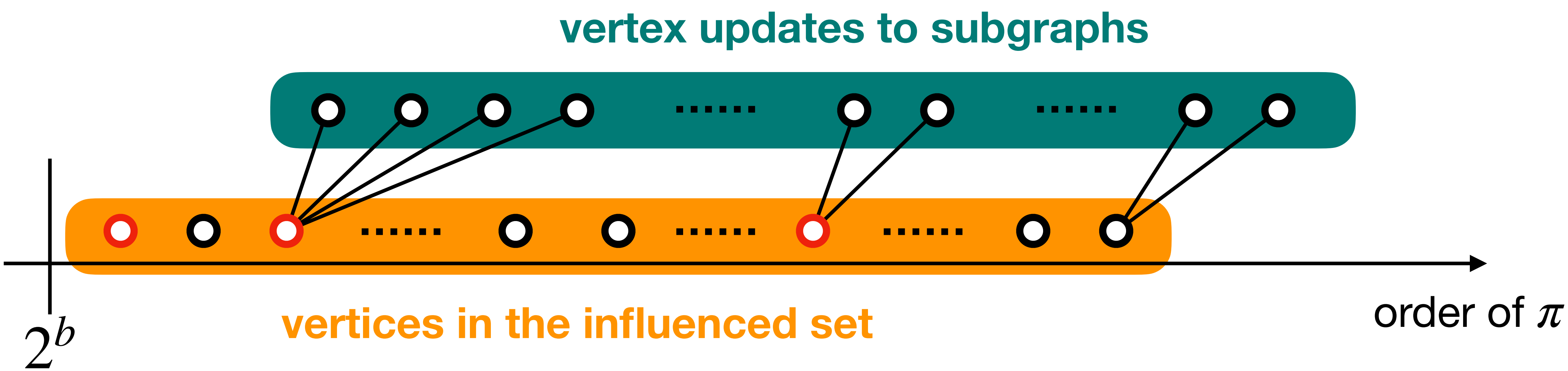


size of [teal box] $\leq$ **max-deg of** G_b *** |S|**

Running time analysis

Say $2^b < \pi(u) < \pi(v) \le 2^{b+1}$, want update time $\le \tilde{O}(n^2/2^{2b})$

Extending [CHK'16], we could prove that $\mathrm{E}_{\pi}[\,|S|\,] \le O(n/2^b)$



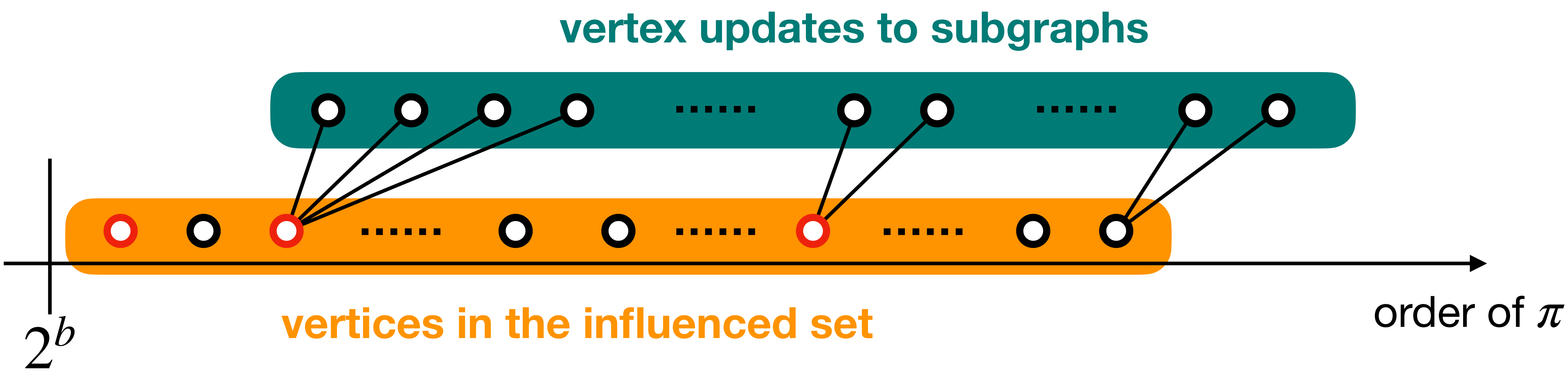
size of  $\le$ **max-deg of** G_b *** |S|**

A trivial bound: update time $\le$ (max-deg of $G_b)^2 \cdot |S| \le \tilde{O}(n^3/2^{3b})$

Running time analysis

Say $2^b < \pi(u) < \pi(v) \leq 2^{b+1}$, want update time $\leq \tilde{O}(n^2/2^{2b})$

Extending [CHK'16], we could prove that $E_\pi[\,|S|\,] \leq O(n/2^b)$

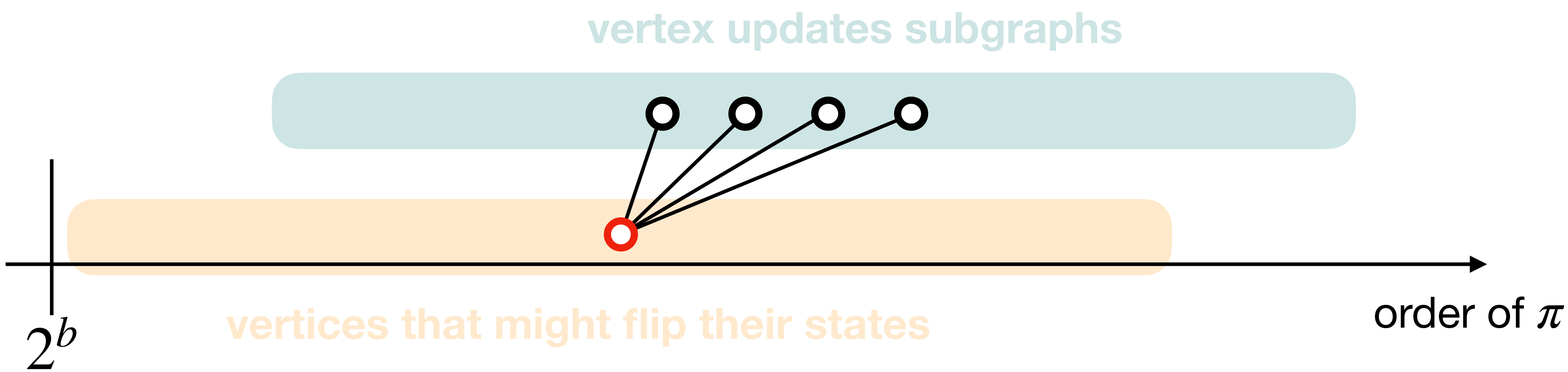


size of [teal box] $\leq$ **max-deg of** G_b *** |S|**

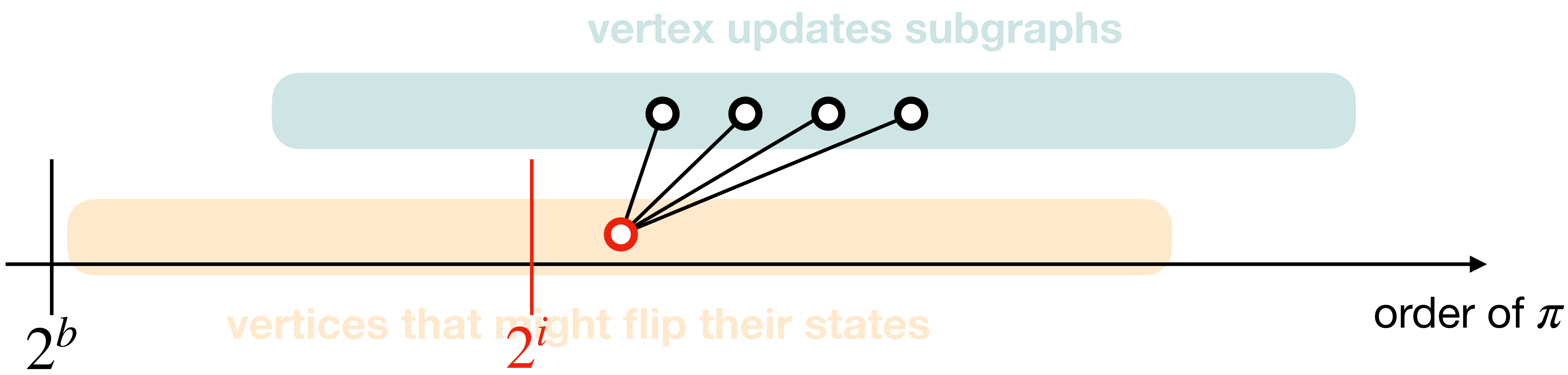
A trivial bound: update time $\leq$ (max-deg of $G_b)^2 \cdot |S| \leq \tilde{O}(n^3/2^{3b})$

We need $\tilde{O}(n^2/2^{2b})$

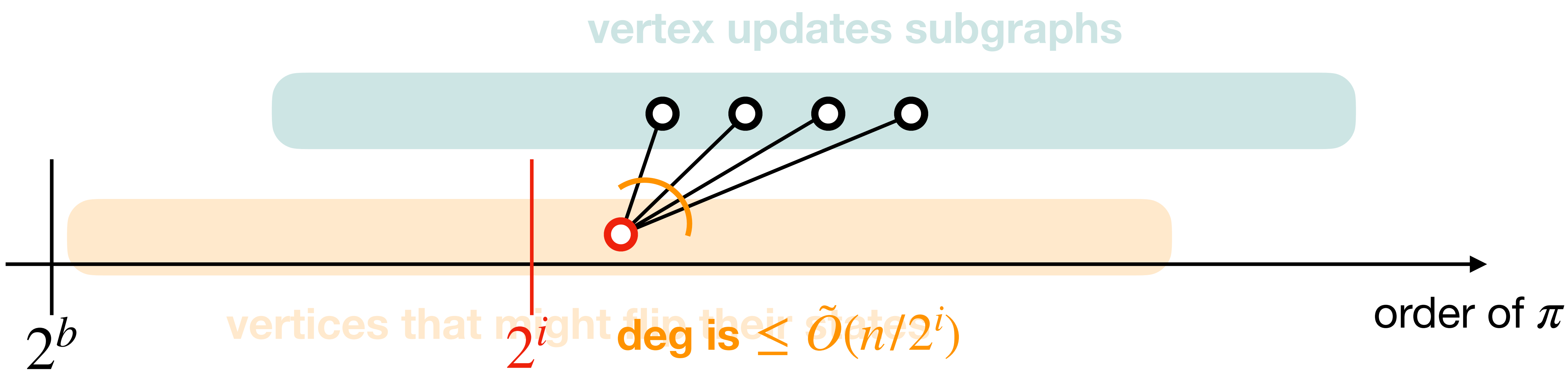
Running time analysis



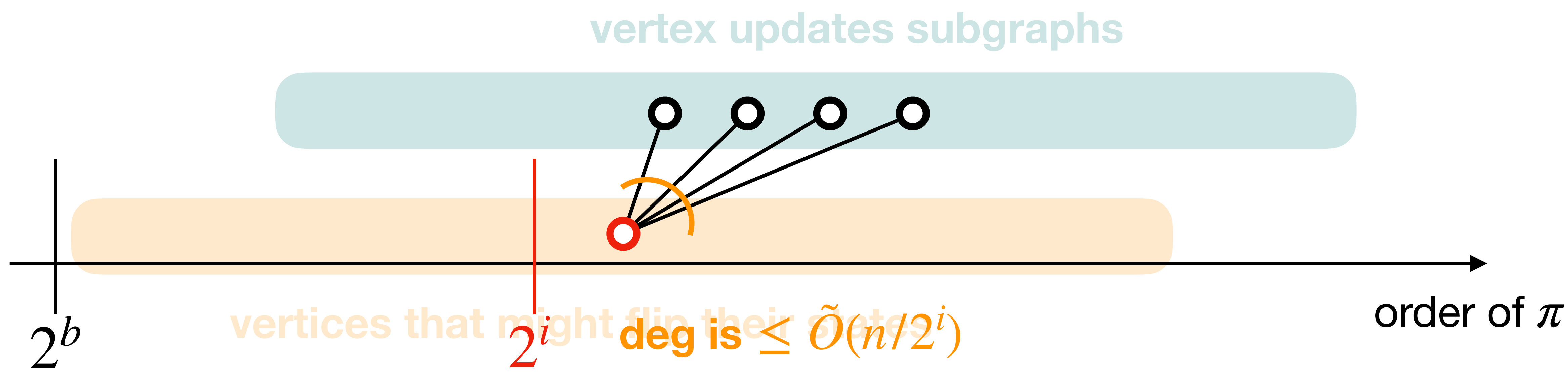
Running time analysis



Running time analysis

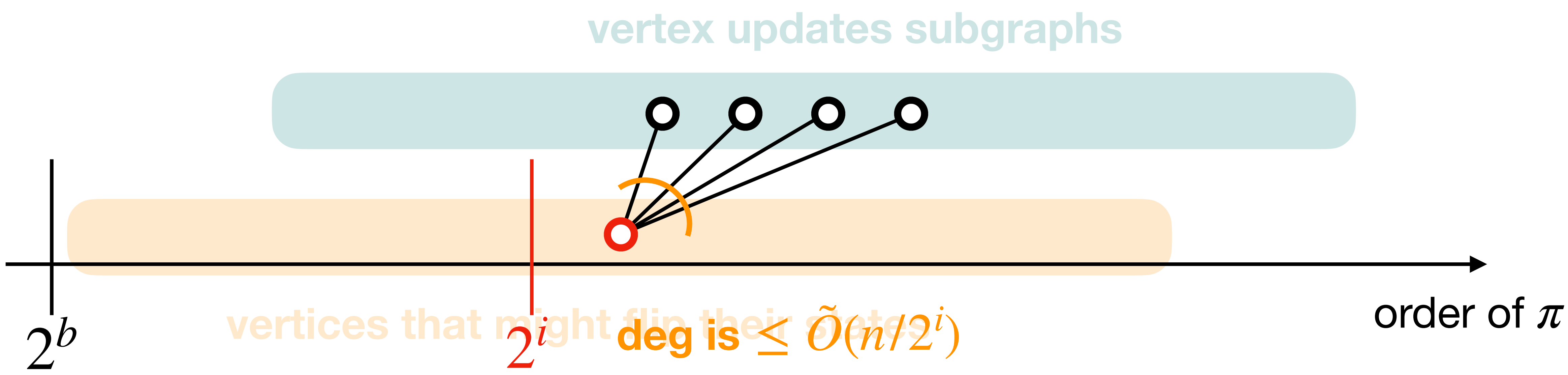


Running time analysis



size of $\leq \sum \tilde{O}(n/2^i) \cdot |S \cap \pi[2^i, 2^{i+1})|$

Running time analysis

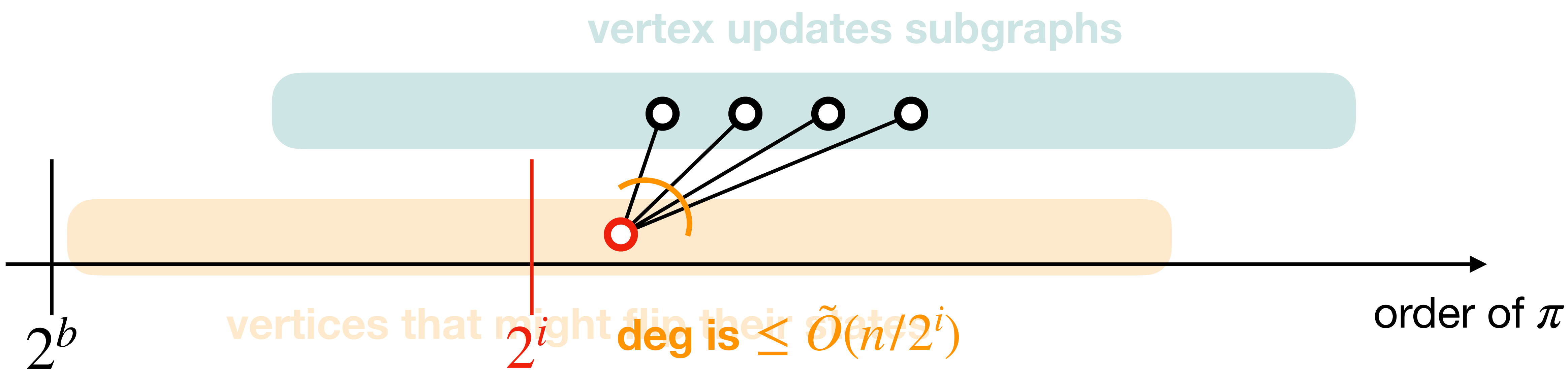


size of $\leq \sum \tilde{O}(n/2^i) \cdot |S \cap \pi[2^i, 2^{i+1})|$

If S is "evenly" distributed on $(2^b, n]$, say $|S \cap \pi[2^i, 2^{i+1})| \leq O(\frac{2^i}{n} \cdot |S|)$

Consequently, $\sum n/2^i \leq$ ~~max deg~~ $O(\log n)$ * |S|

Running time analysis



size of [teal shape] $\leq \sum \tilde{O}(n/2^i) \cdot |S \cap \pi[2^i, 2^{i+1})|$

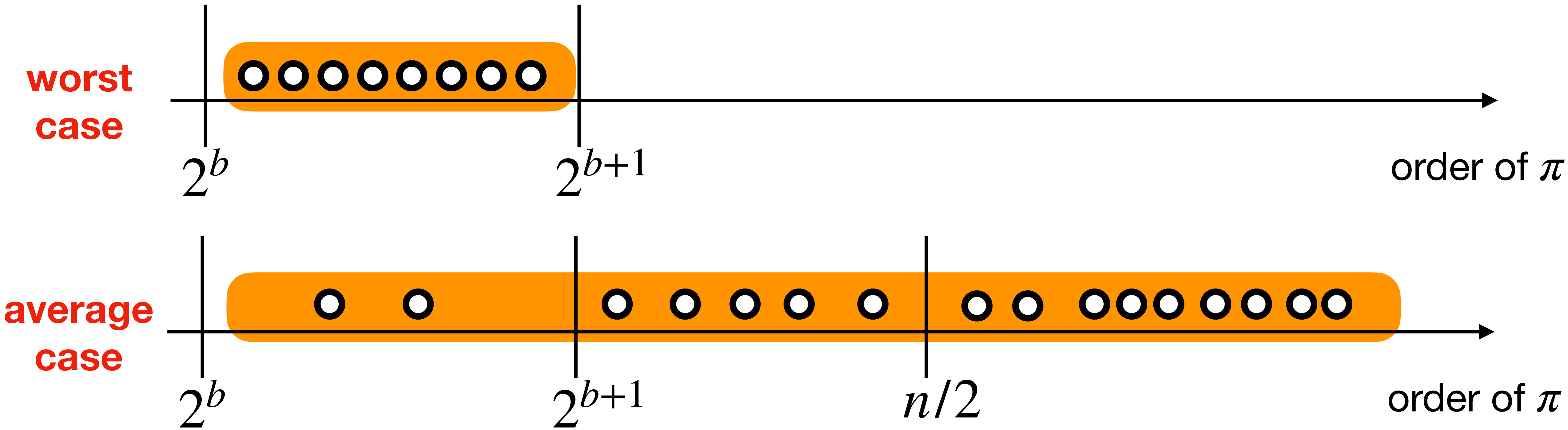
If S is "evenly" distributed on $(2^b, n]$, say $|S \cap \pi[2^i, 2^{i+1})| \leq O(\frac{2^i}{n} \cdot |S|)$

Consequently, $\sum n/2^i \leq$ ~~max-deg~~ $O(\log n)$ * |S|

Total update time becomes $\tilde{O}(n^2/2^{2b})$

Running time analysis

Analyze the how S is distributed on $(2^b, n]$



<u>Lemma:</u>

Conditioning on $2^b < \pi(u) < \pi(v) \leq 2^{b+1}$, the set S is "evenly" distributed on interval $(\pi(v), n]$

Running time analysis

Running time analysis

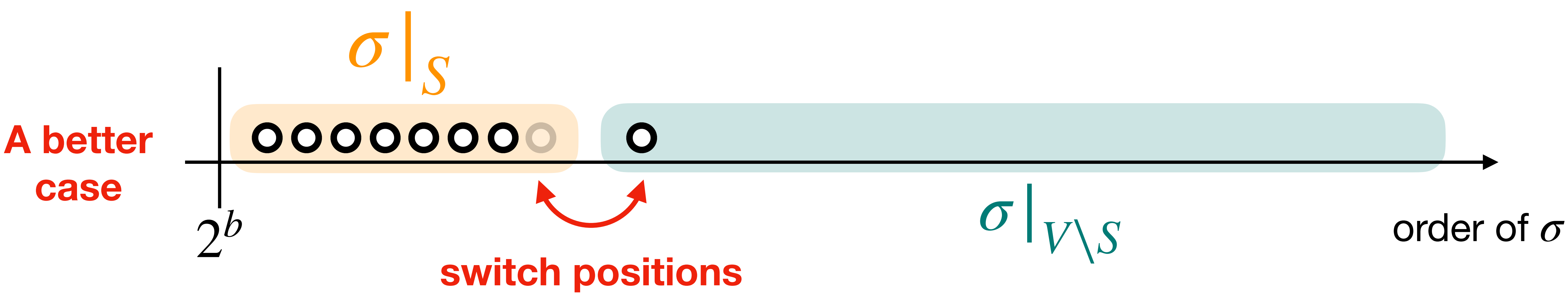
Strategy:

Gradually transform the bad case to a good case.

Running time analysis

Strategy:

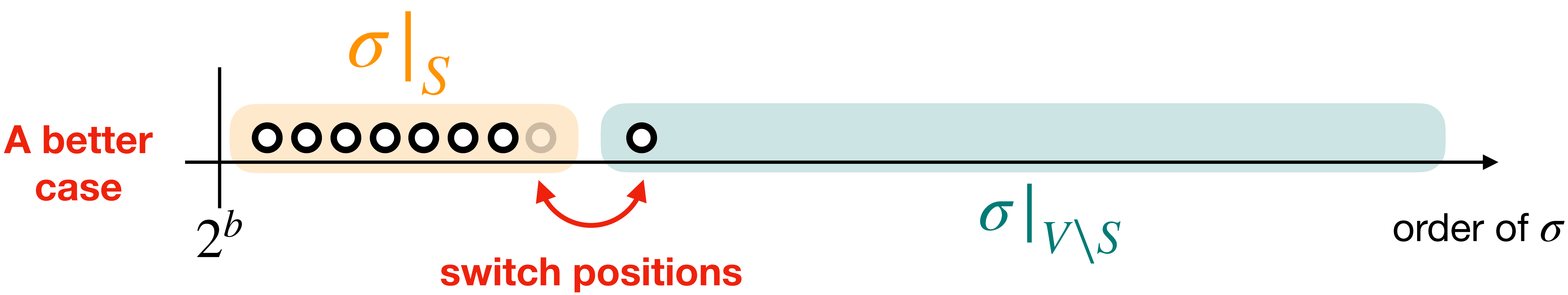
Gradually transform the bad case to a good case.



Running time analysis

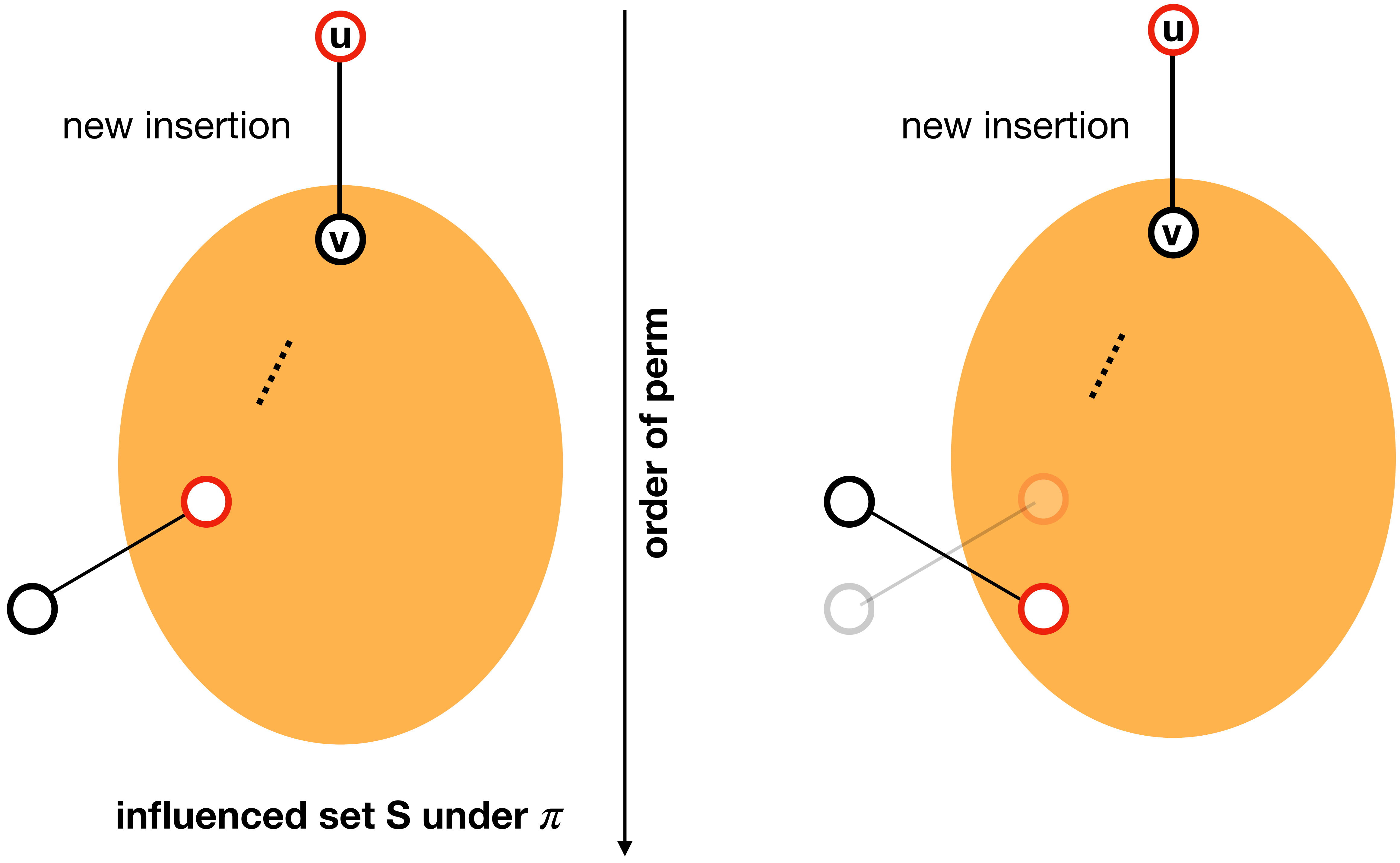
Strategy:

Gradually transform the bad case to a good case.

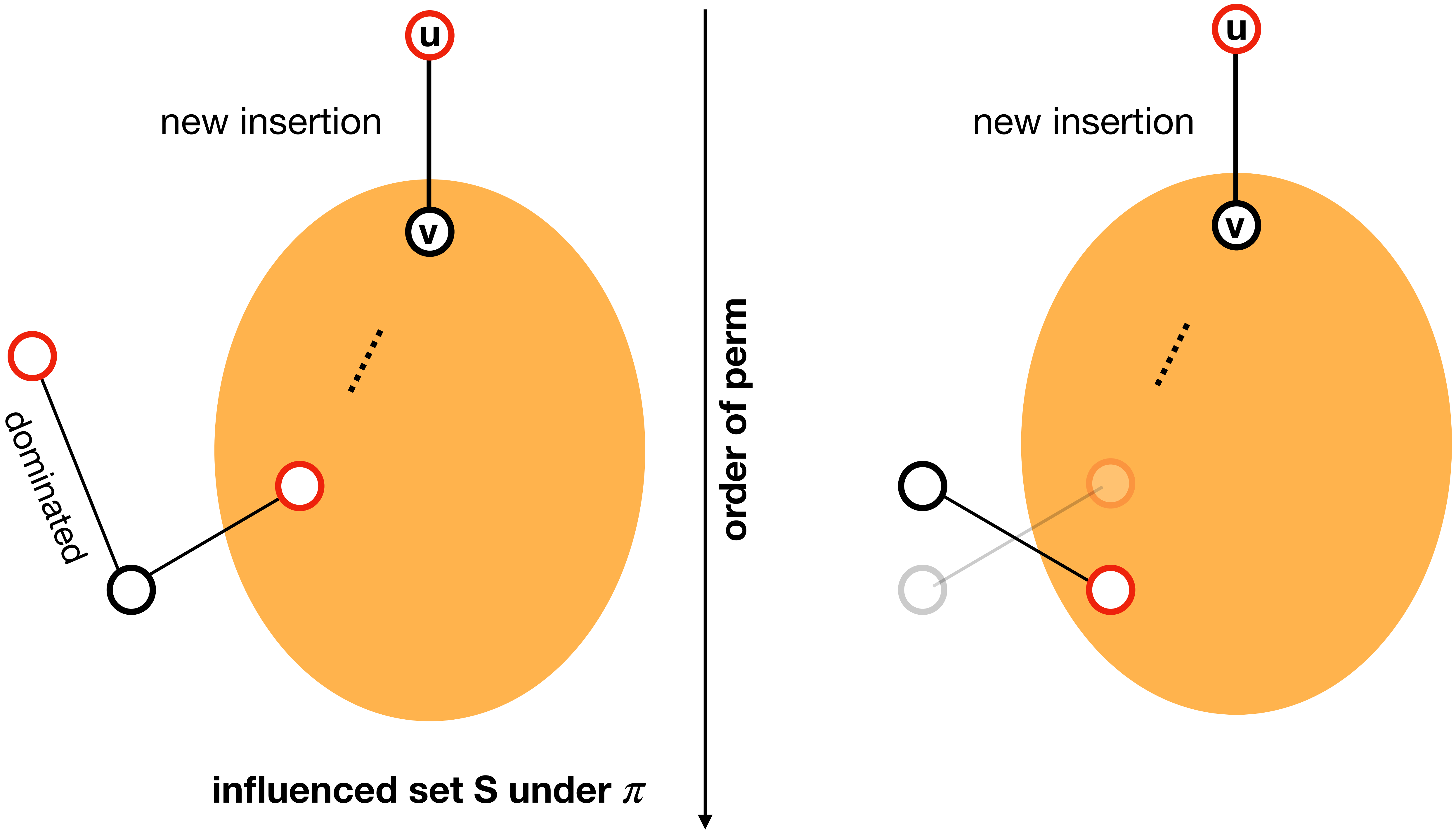


Prove S is still the influenced set under the new permutation σ

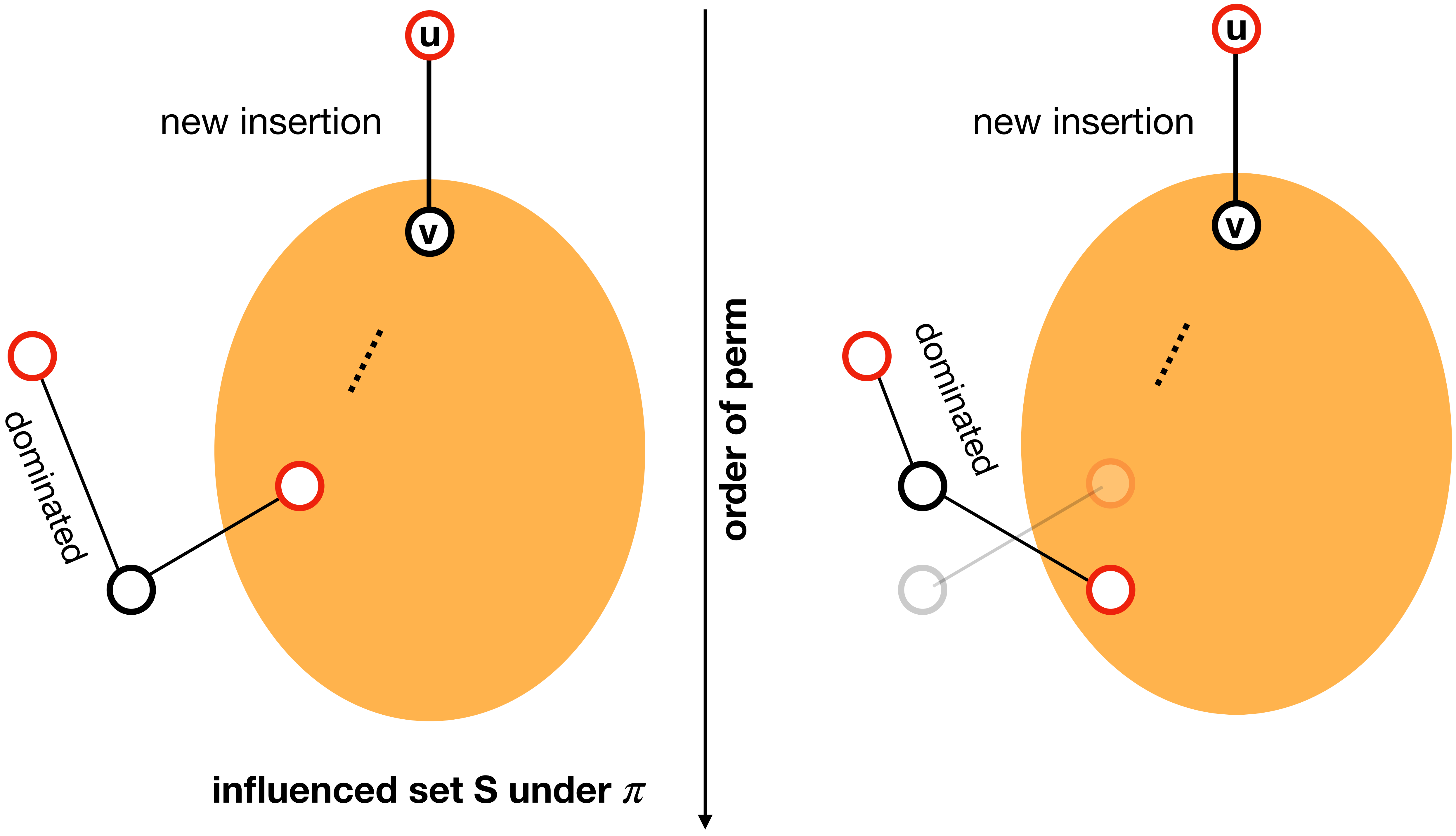
Running time analysis



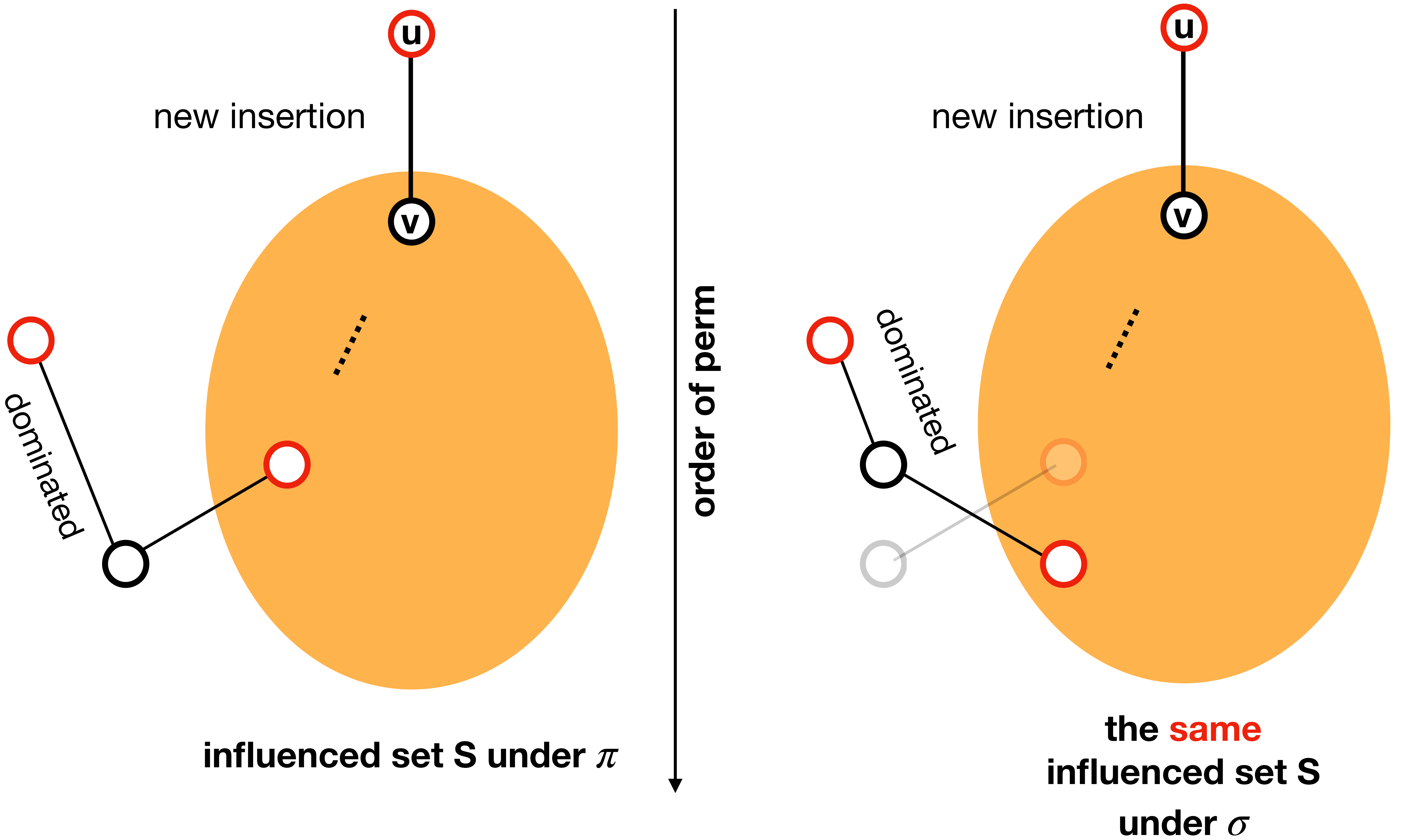
Running time analysis



Running time analysis



Running time analysis



Thanks