Chapter 2.

# OMv Lower Bounds

Danupon Nanongkai

**KTH, Sweden**

ADFOCS 2018
Last edited: Aug. 15, 2018

Part 1

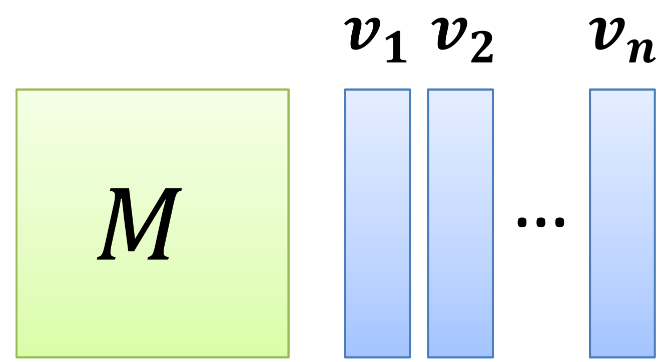
# THE CONJECTURES

# OMv Conjecture

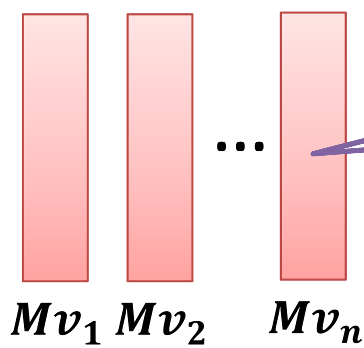
(**O**nline **M**atrix-**V**ector Multiplication) [Henzinger, Krinninger, N, Saranurak, STOC'15]

**Input:** $n \times n$ Boolean matrix **M**

**Then:** $n$ Boolean vectors $v_i$




**Output:**




(OR,AND)-mult.
**not** (+,×)-mult.

Answer $Mv_i$ before getting $v_{i+1}$

**Conjecture:** No algorithms with **total** time $O(n^{3-\epsilon})$

**Current Best:** $O(n^3/2^{\sqrt{\log n}})$ [Larsen-Williams SODA'17]

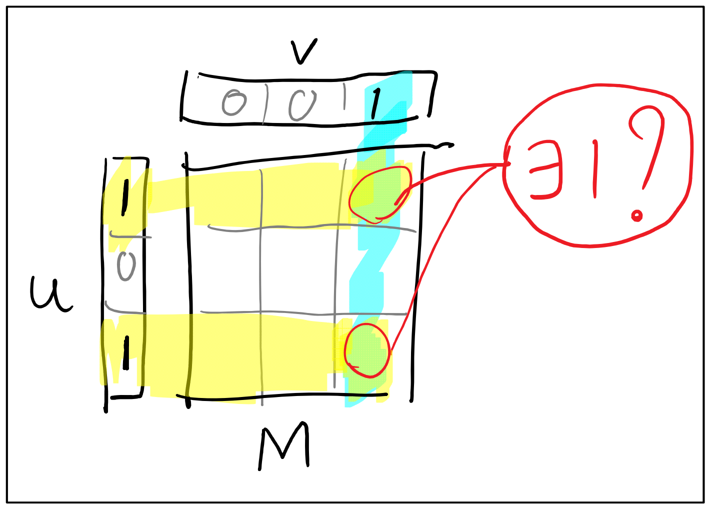
Example on board?

# OuMv Conjecture (Matrix Form)

**Input:** $n \times n$ Boolean matrix **M**

**Then:** $n$ **pairs** of Boolean vectors $(u_i, v_i)$

**Output:** $u_i^T M v_i$




Answer $u_i^T M v_i$ before getting $(u_{i+1}, v_{i+1})$

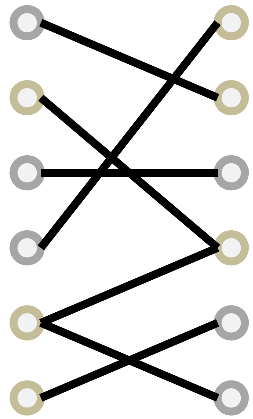
**Conjecture:** No algorithms with **total** time $O(n^{3-\epsilon})$

***even with polynomial time to process M!***

Example on board?

# OuMv as Independent set

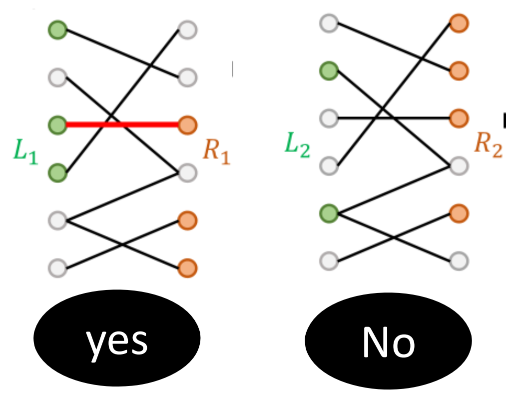
**Preprocess:**



$poly(n)$ time
e.g. $n^{100}$

Output before next input arrives

**Input:** $(L_1, R_1)$
…
$(L_n, R_n)$




OuMv Conj → No $n^{3-\epsilon}$ time

**Output:** Any edge linking $L_1$ and $R_1$?
…
Any edge linking $L_n$ and $R_n$?

$L_1 \cup R_1$ is independent set?

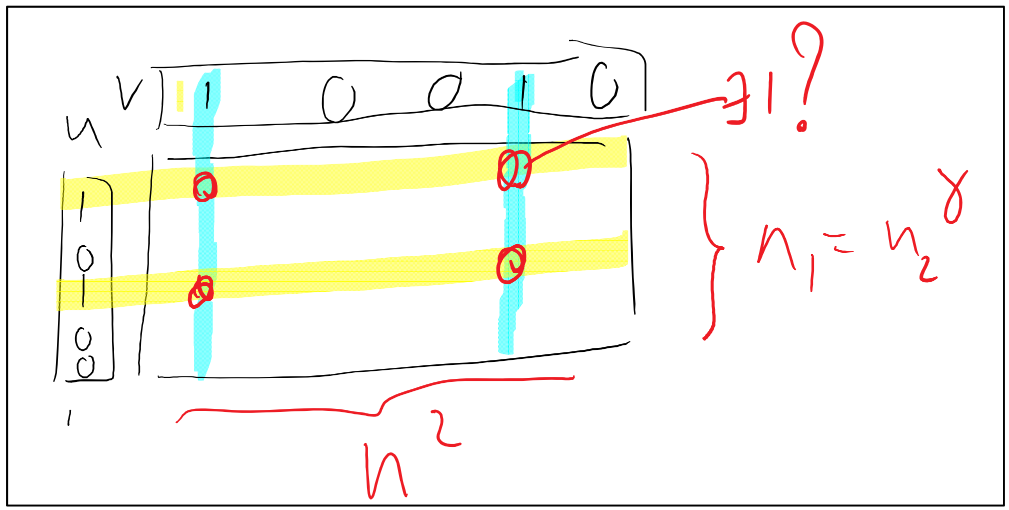
Write on board

# $\gamma$-OuMv Conjecture (or just a "free-form" of OuMv)

**Input:** $n_1 \times n_2$ Boolean matrix **M**, $n_1 = \boldsymbol{n_2^{\gamma}}, \gamma > 0$.

**Then:** $n_3$ **pairs** of Boolean vectors $(\boldsymbol{u_i}, \boldsymbol{v_i})$

**Output:** $u_i^T M v_i$




**Conjecture:** No algorithms with **total** time $O((n_1 n_2 n_3)^{1-\epsilon})$

***even with polynomial time to process M!***

Example on board?

# Formal Statements

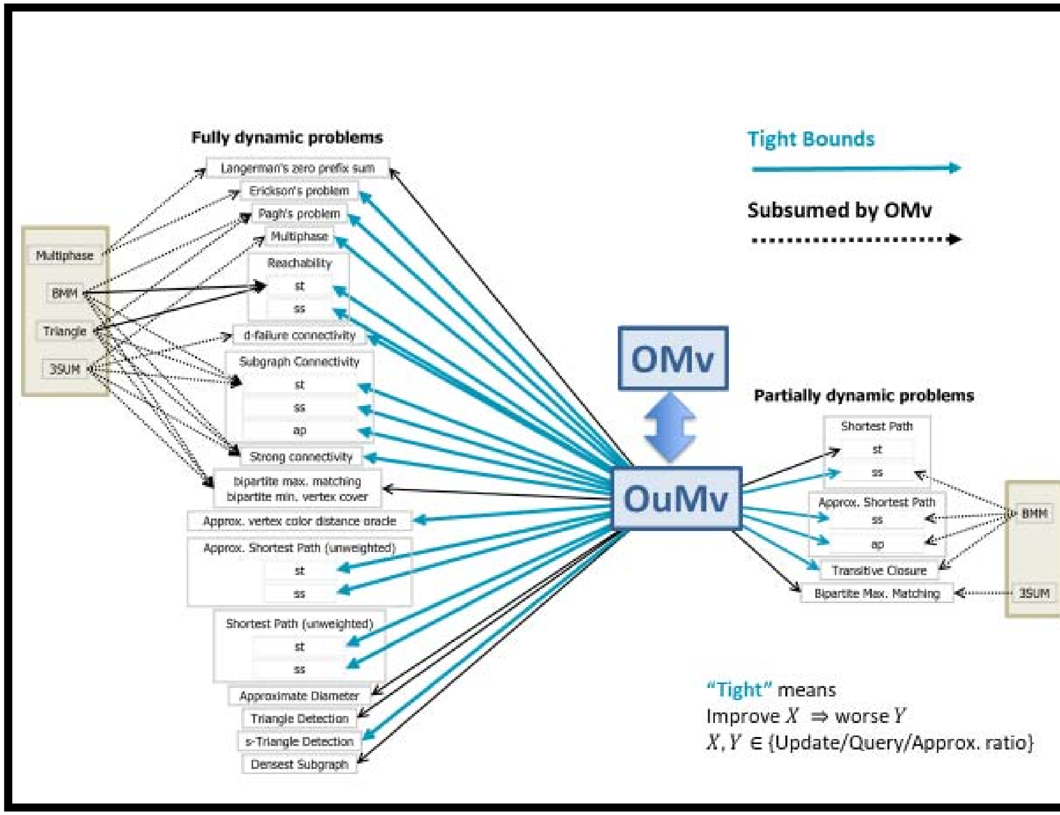
<u>OMv Conjecture</u>: For any constant $\epsilon > 0$, there is no $O(n^{3-\epsilon})$-time algorithm that solves OMv with an error probability of at most 1/3.

<u>$\boldsymbol{\gamma}$-OuMv Conjecture</u>: For any constant $\gamma > 0, \epsilon > 0$, there is no algorithm for $\boldsymbol{\gamma}$-OuMv with parameters $n_1, n_2, n_3$ using preprocessing time $poly(n_1, n_2)$ and computational time $O((n_1 n_2 n_3)^{1-\epsilon})$ that has error probability of at most 1/3.

# **Theorem**: OMv **implies** $\gamma$-OuMv

**<u>Plan</u>**

- Some lower bounds from OuMv
- Prove above Theorem



Part 2

# Some Update Time Bounds

# Example set 1: Fully-Dynamic Graphs

After each edge insertions/deletion check:

1. st-reachability
2. undirected st-shortest paths
   – Unweighted/weighted
3. strong edge-connectivity

**These bounds hold against amortization & randomization!**

Main reason: $\gamma$-OuMv allows arbitrary (polynomial) preprocessing time and number of updates.



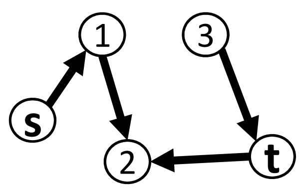
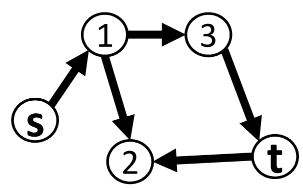
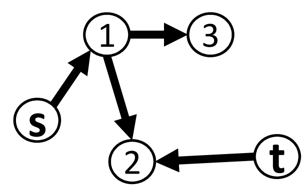
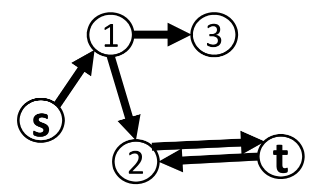
$n$ = # of nodes, $m$=# of edges

What lower bound can you prove under OMv?

Example 1.1

# st-Reachability

# Dynamic st-Reachability Problem

| **Input:** Update in G | | insert(1,3) | delete(3,t) | insert(2,t) |
|---|---|---|---|---|
| **Picture** |  |  |  |  |
| **Output:** s reach t? | No | Yes | No | Yes |



Thanks Thatchaphol Saranurak for slides

# Known Results for st-Reach

- Incremental: **O(1) amortized** update time
- **$\boldsymbol{\Omega(n)}$ lower bound assuming OuMv**
  - Hold against randomized and amortized algorithms
  - … even with oblivious-adversary & empty-start assumptions
  - Higher lower bound for a related problem called #SSR
  - $\Omega(n^2)$ lower bound for "combinatorial" algorithms
- Fully-dynamic: $\boldsymbol{\Theta(n^{1.407})}$ **worst-case** update time
  - Lower bound assumes a variant of OuMv



$n$ = # of nodes, $m$=# of edges

# Will show…

**st-Reach**
- Preprocess: $poly(n)$
- Update: $n^{1-\epsilon}$ (amortized)

So this cannot exist

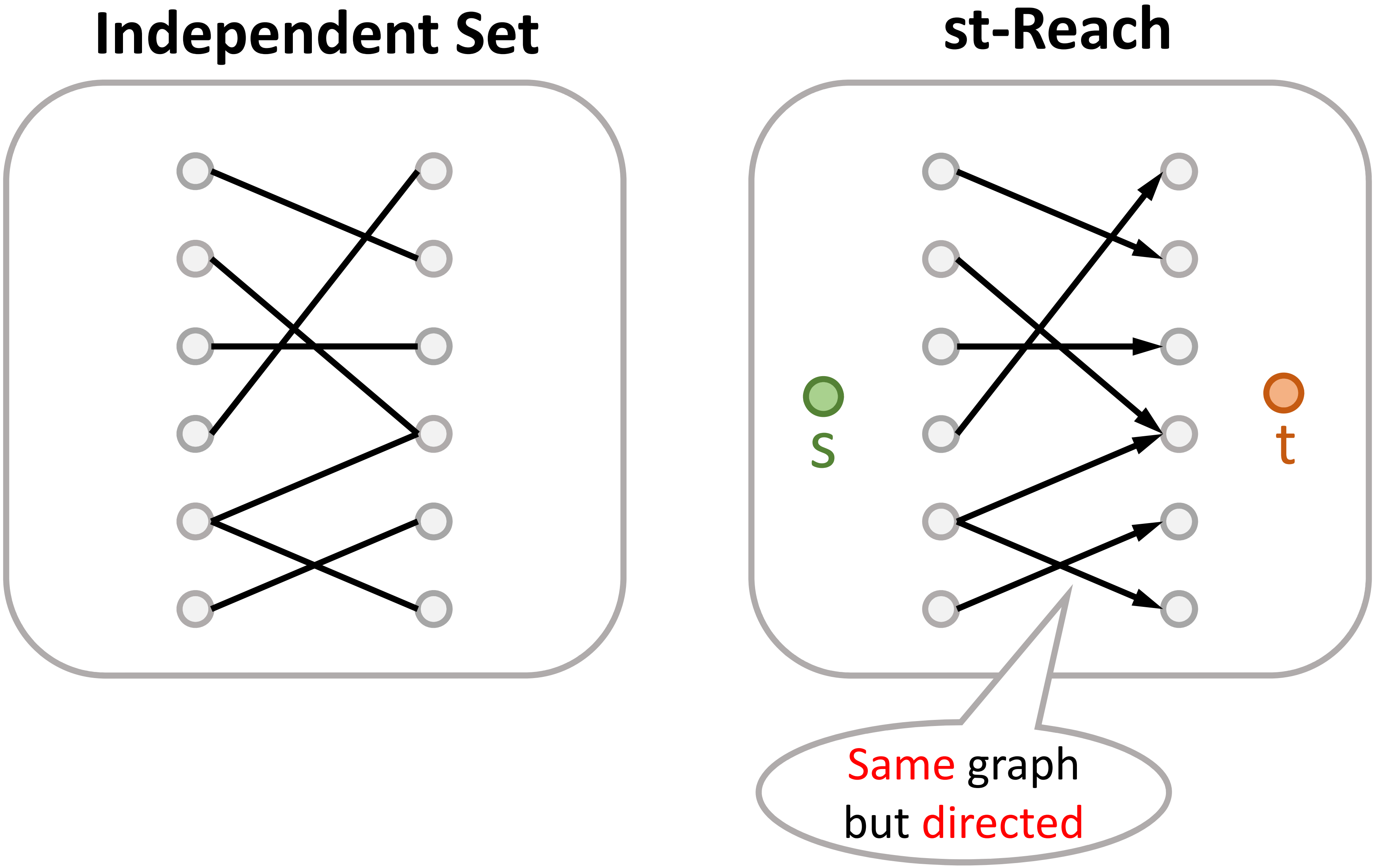
**Independent Set**
- Preprocess: $poly(n)$
- Time (for $n$ queries): $n^{3-\epsilon}$

Impossible!
assuming OMv

$n$ = # of nodes, $m$=# of edges

Thanks Thatchaphol Saranurak for slides

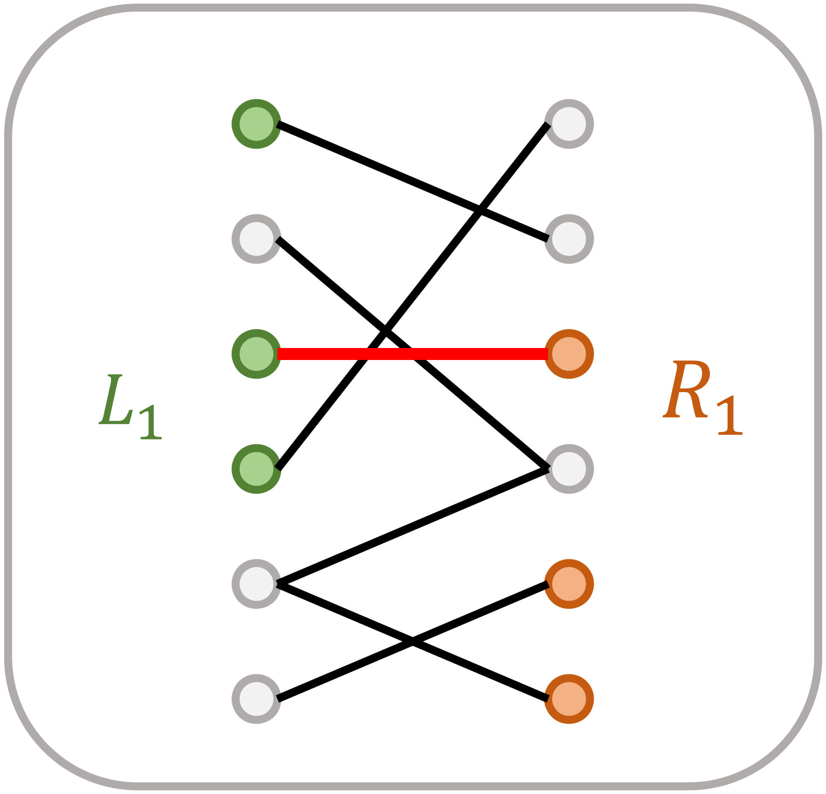
# Preprocess



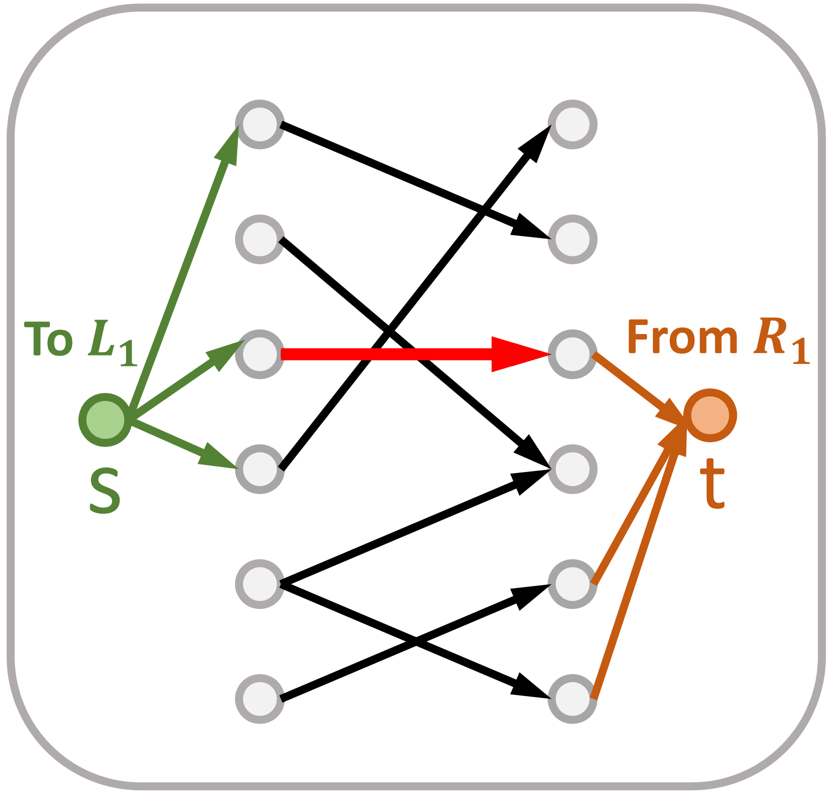

Thanks Thatchaphol Saranurak for slides

# Edge($L_1$, $R_1$)?

## Independent Set




## st-Reach



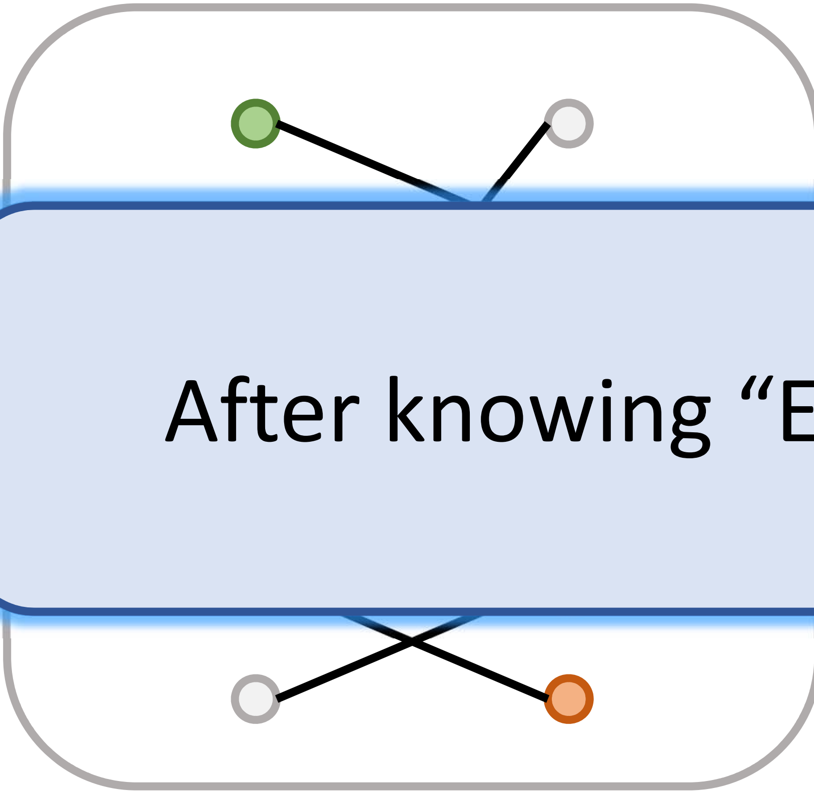

After O(n) updates…

∃ an edge linking $L_1$ and $R_1$ ⟺ s can reach t

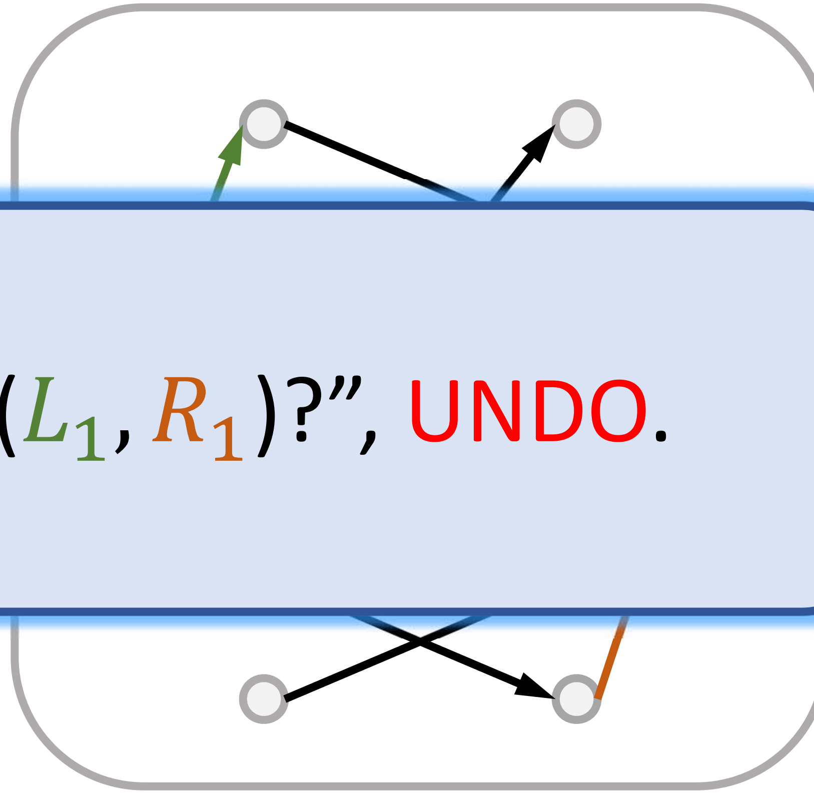
Thanks Thatchaphol Saranurak for slides

Edge($L_1$, $R_1$)?

**Independent Set**



**st-Reach**

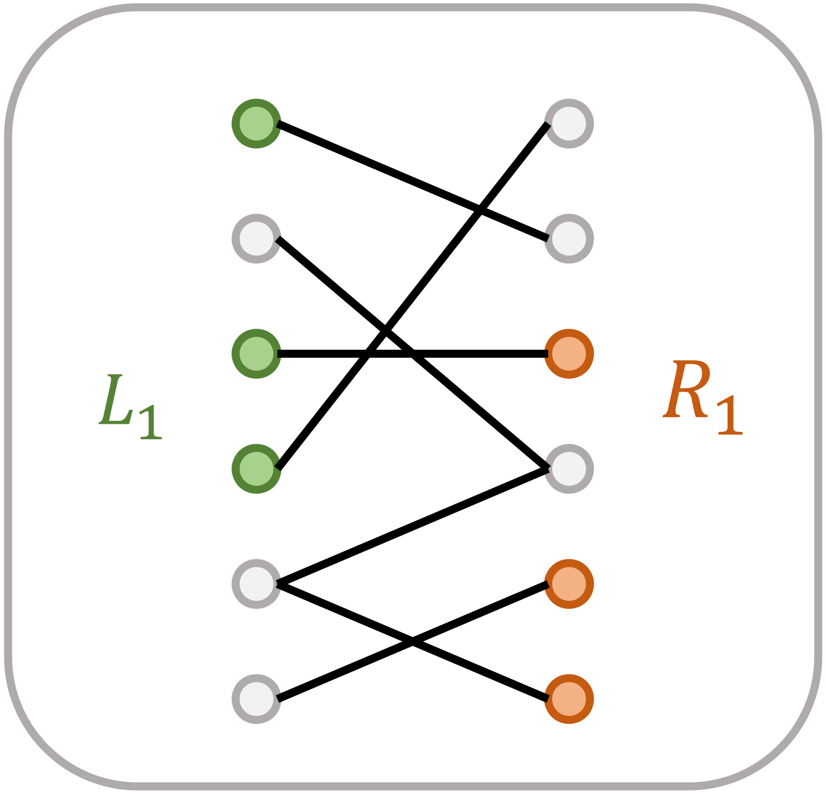


After knowing "Edge($L_1$, $R_1$)?", UNDO.

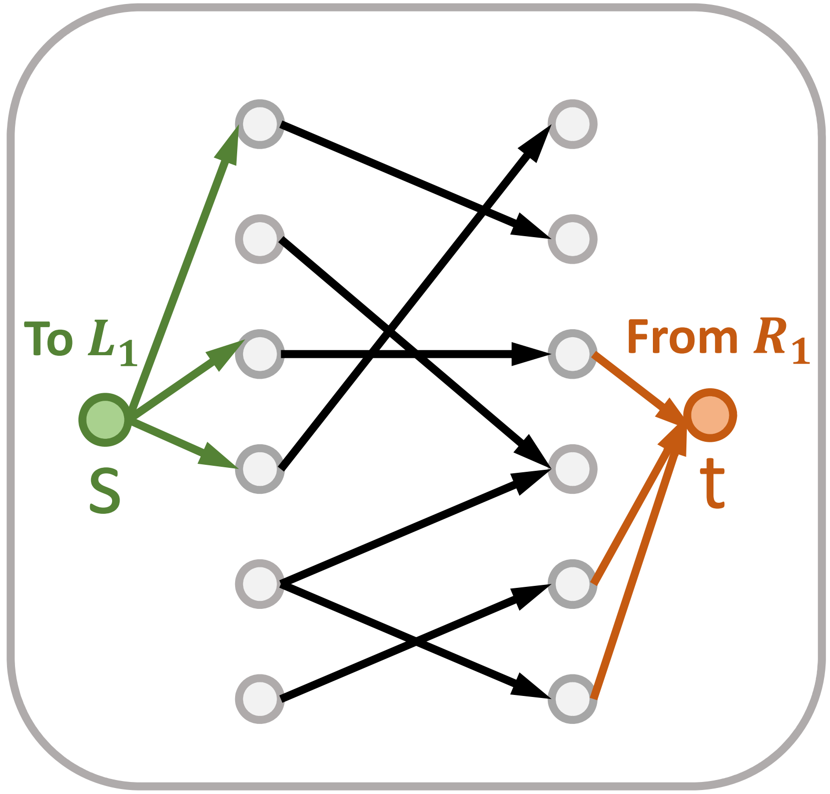
Thanks Thatchaphol Saranurak for slides

# Edge($L_1$, $R_1$)?

**Independent Set**




**st-Reach**

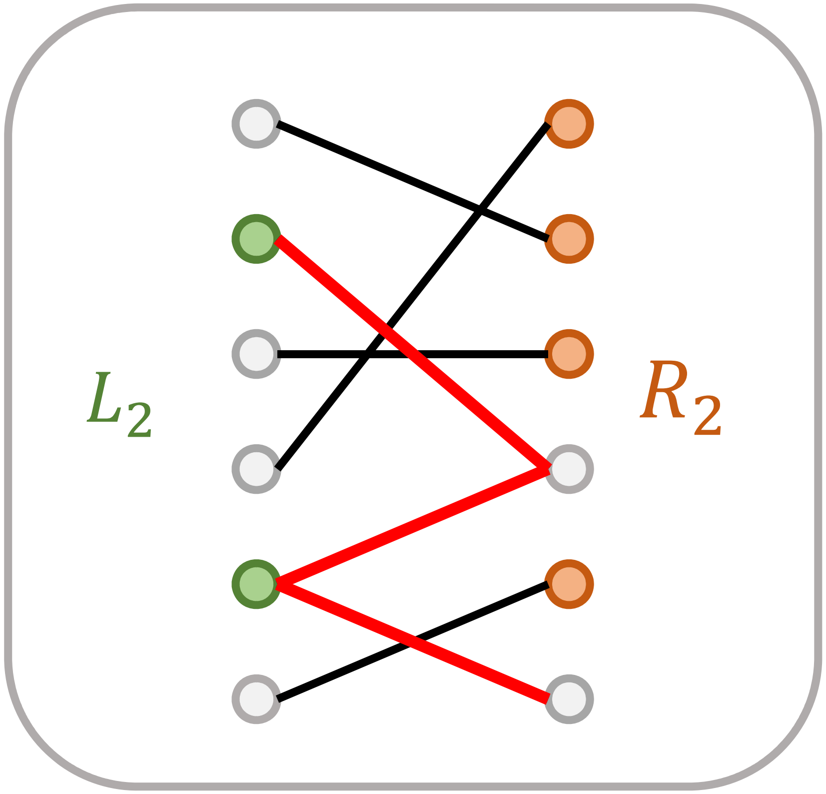



Use O(n) updates.

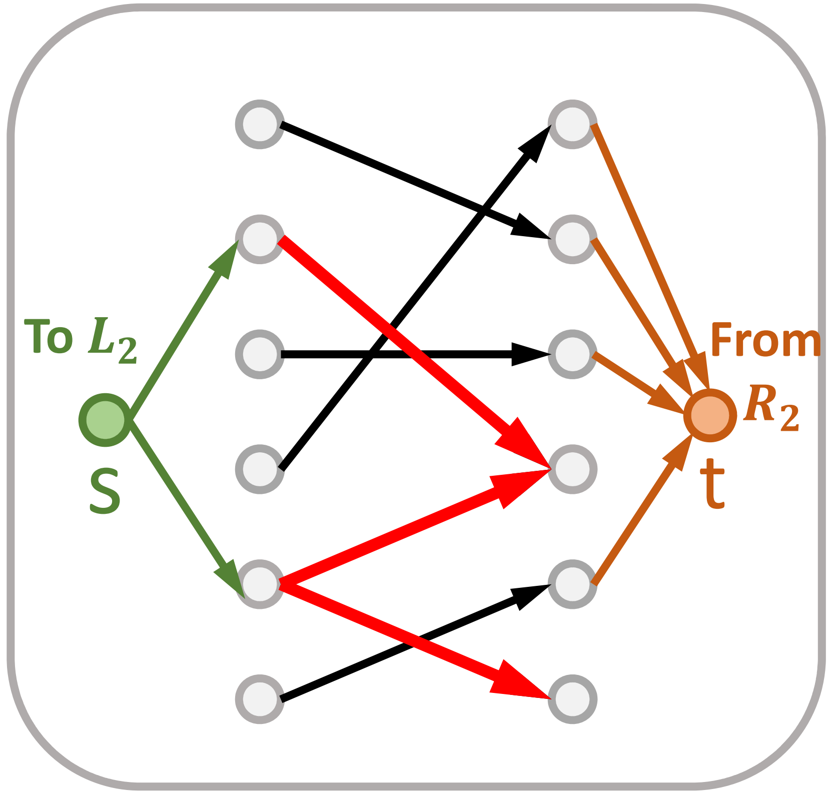
Thanks Thatchaphol Saranurak for slides

# Edge($L_2$, $R_2$)? (another example)

## Independent Set




## st-Reach




Not ∃ an edge linking $L_2$ and $R_2$ ⟺ After O(n) updates… s can not reach t

Thanks Thatchaphol Saranurak for slides

# Check: The lower bound hold for amortized update time?

- Suppose that an algorithm $\boldsymbol{A}$ for st-reach takes $\boldsymbol{O}(\boldsymbol{n}^{0.9}\boldsymbol{t})$ time after $\boldsymbol{t}$ updates, when start from an empty graph.
- Setting up the original bipartite graph: Take $\boldsymbol{O}(\boldsymbol{n}^{2.9})$ time to insert $\boldsymbol{n}^{2}$ edges.
- Handling one pair of $(u_i, v_i)$: Take $\boldsymbol{O}(\boldsymbol{n}^{1.9})$ time to insert $\boldsymbol{n}$ edges.

→ Take $\boldsymbol{O}(\boldsymbol{n}^{2.9})$ time to handle all pairs of vectors

# Check: The lower bound hold against randomized algorithms?

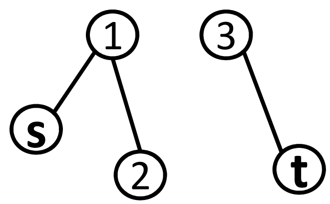
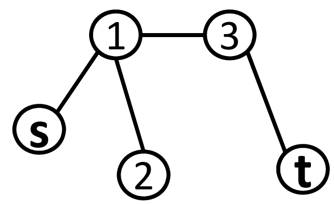
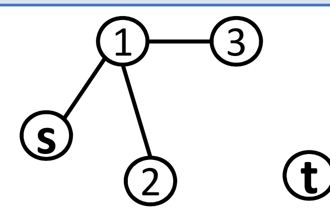
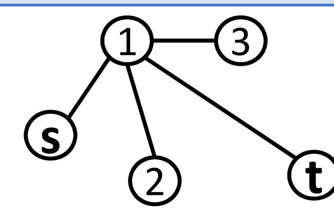
- The conjecture was also for randomized algorithms.
- The reduction is between decision problems. There is no difference between oblivious and non-oblivious adversary.
  - Must be more careful for, e.g. approximation algorithms.

Example 1.2

# st-Distance

(Undirected)

# Dynamic st-Distance Problem

| **Input:** Update in G | | insert(1,3) | delete(3,t) | insert(1,t) |
|---|---|---|---|---|
| **Picture** |  |  |  |  |
| **Output:** st-distance | $\infty$ | 3 | $\infty$ | 2 |

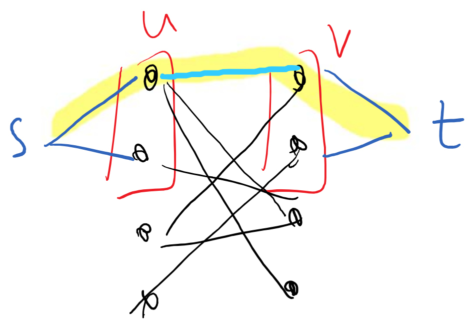
- Easy: $\Omega(n)$ lower bound for exact version
- How about approximate version?

# $\Omega(n)$ for unweighted (5/3-$\epsilon$)-approximation

Output number $x$ s.t.

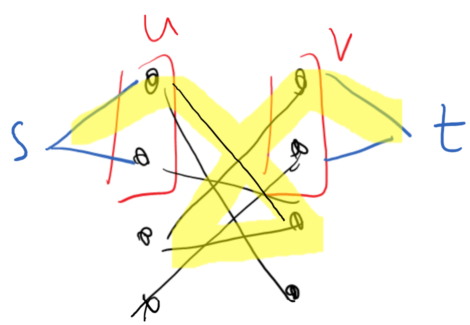
$$dist(s,t) \leq x \leq \left(\frac{5}{3} - \epsilon\right) dist(s,t)$$

**Same reduction as st-Reachability**




uMv = 1 ➔ dist(s,t)=3

Algorithm's output $\leq \left(\frac{5}{3} - \epsilon\right) 3 < 5$




uMv = 0 ➔ dist(s,t)≥5

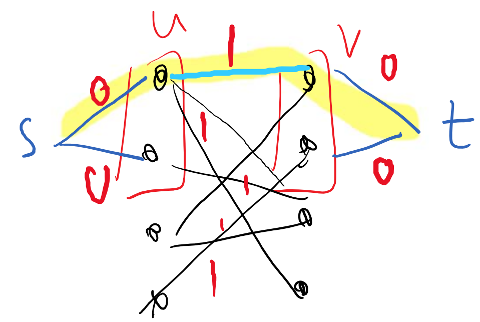
Algorithm's output $\geq 5$



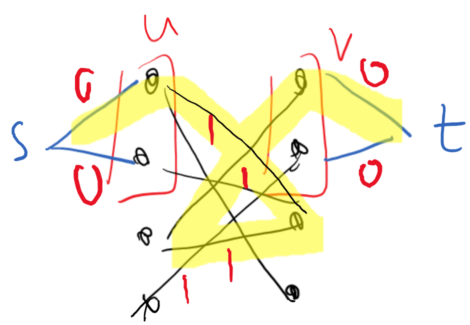
$n$ = # of nodes, $m$=# of edges

# $\Omega(n)$ for weighted (3-$\epsilon$)-approximation

**Same reduction as st-Reachability**




uMv = 1 ➔ dist(s,t)=1




uMv = 0 ➔ dist(s,t)≥ 3



$n$ = # of nodes, $m$=# of edges

# Known Results

**Fully-dynamic**

- $\boldsymbol{\Omega(n)}$ **lower bound assuming OuMv** for (5/3-$\epsilon$)-approx
  - Hold against randomized and amortized algorithms
  - … even with oblivious-adversary & empty-start assumptions
  - **Hold against (small-)approximation algorithms**
- $\mathbf{O}(\boldsymbol{n}^{1.724})$ **worst-case** update time for $(\mathbf{1}+\boldsymbol{\epsilon})$**-approx**

**Incremental/decremental:**

- **Exact:** $\Theta$(n) amortized update time, $\Theta(m)$ worst-case
- $(\mathbf{1}+\boldsymbol{\epsilon})$**-approx:** $O\left(n^{o(1)}\right)$ amortized



$n$ = # of nodes, $m$=# of edges

Example 1.3

# Strong Edge-Connectivity

# Dynamic Strong Edge-Connectivity Problem

| Input | Update | Output |
| --- | --- | --- |
| A directed graph | Edge insertions/deletions | Is the graph strongly connected?<br>(Every s can reach every t) |

# $\Omega(n)$ for strong edge-connectivity

- Reduce from st-Reachability by adding
  - edges $E_1$ from **t** to every node, and
  - edges $E_2$ from every node to **s.**
- Observe: Adding edges pointing to s and from t does not change st-reachability.

- If **t** is **not** reachable from **s**, this remains the case.
- If **t** is reachable from **s**, then
  - **s** can reach all nodes via $E_1$, and
  - all nodes can reach **s** via $E_2$

- Easy: Extend to $\Omega(\sqrt{m})$ lower bound

# Example set 2: Non-Graph Problems

1. Erickson's Problem
2. Pagh's Problem

**These bounds hold against amortization & randomization!**



$n$ = # of nodes, $m$=# of edges

What lower bound can you prove under OMv?

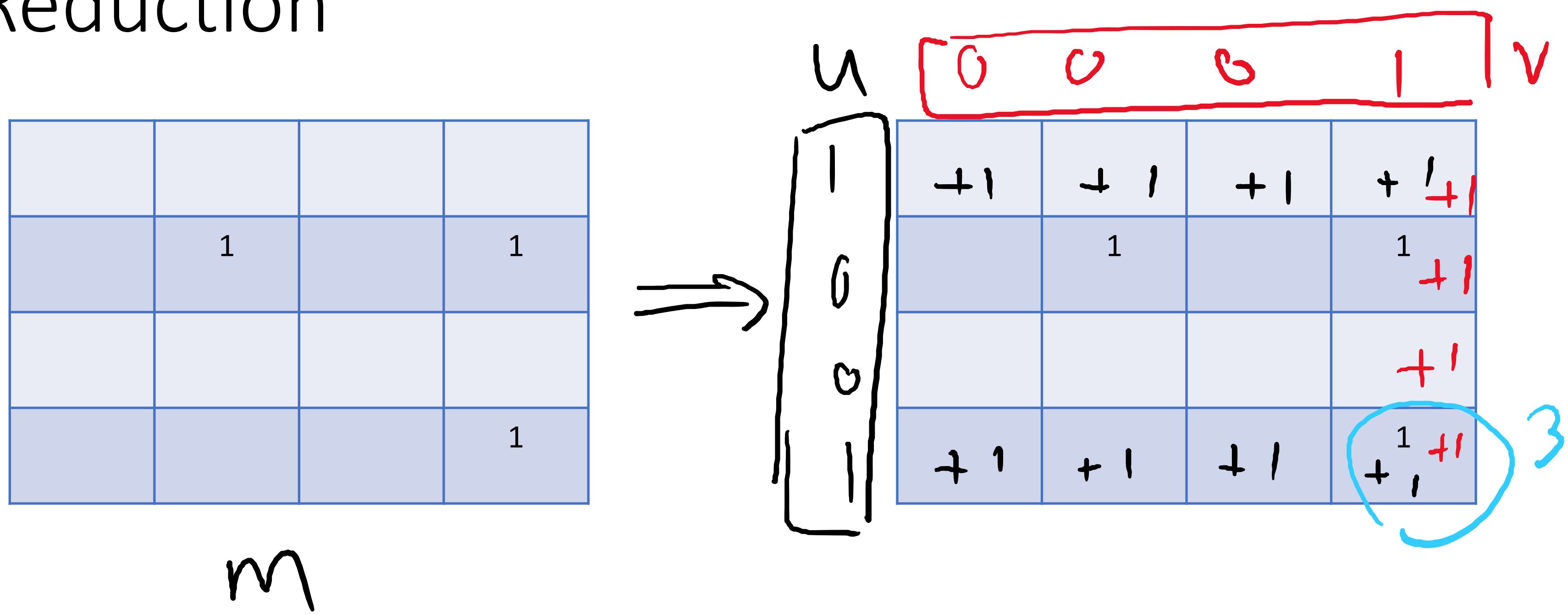
Example 2.1

# Erickson’s problem

# Erickson's problem

| Name | Input | Update | Query |
|---|---|---|---|
| Erickson's Problem | A matrix of integers of size $n \times n$ | Increment all values in a specified row or column | Find the maximum value in the matrix |

# Reduction



Example 2.2

# Pagh’s problem

# Pagh's problem (a variant)

- **Input:** $k$ subsets $X_1, X_2, \ldots, X_k$ over a universe $U = \{1, \ldots, k\}$
- **Update:** Given a pointer to two subsets $X_i$ and $X_j$, create a new subset $X_i \cap X_j$
- **Output:** After each update outputs whether the new subset is empty or not.

# Pagh's problem -- Reduction

| | | | |
|---|---|---|---|
| | | | |
| | 1 | | 1 |
| | | | |
| | | | 1 |

$m$

| | | | |
|---|---|---|---|
| | | | |
| | 0 | | 0 |
| | | | |
| | | | 0 |

$\bar{m}$

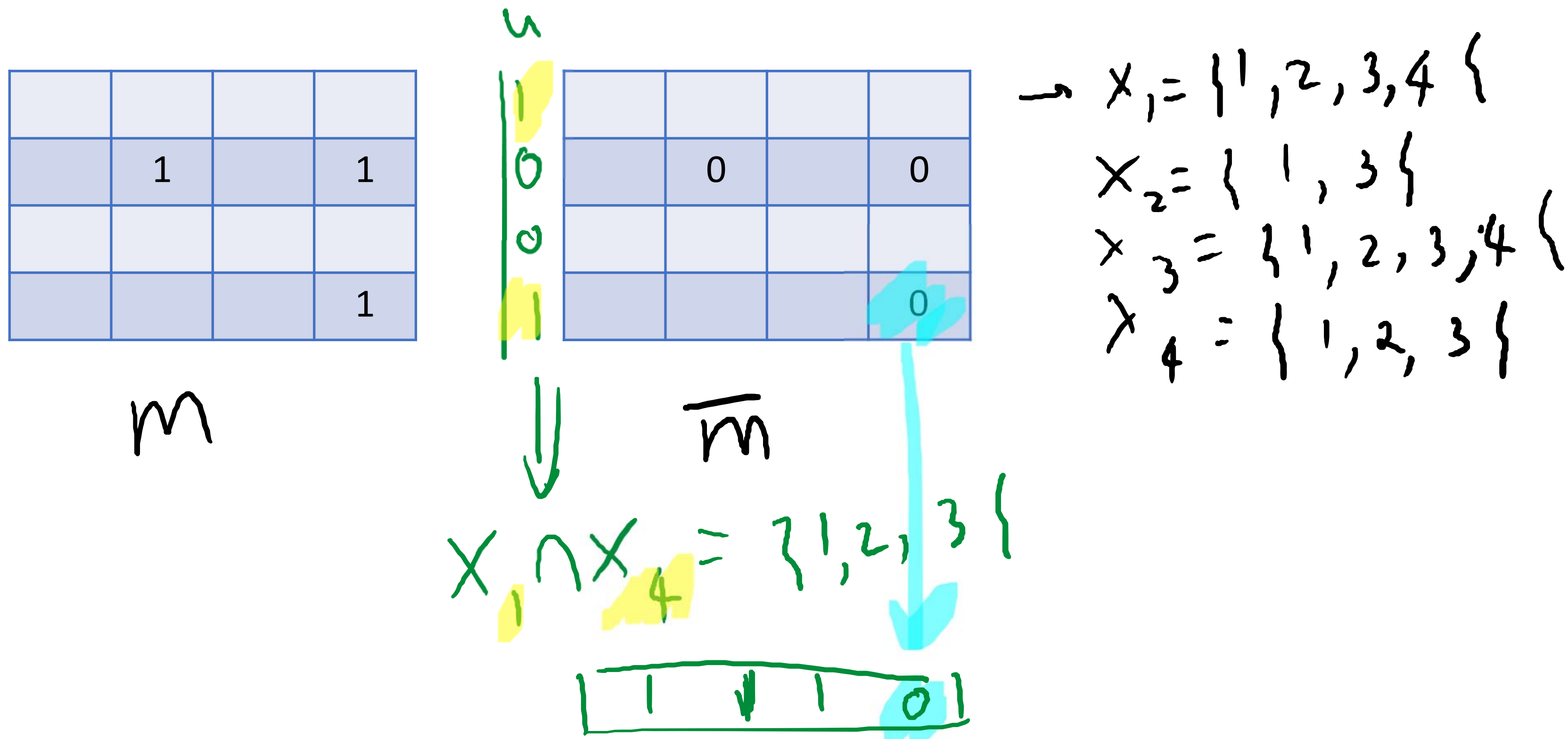
→ $x_1 = \{1, 2, 3, 4\}$

$x_2 = \{1, 3\}$

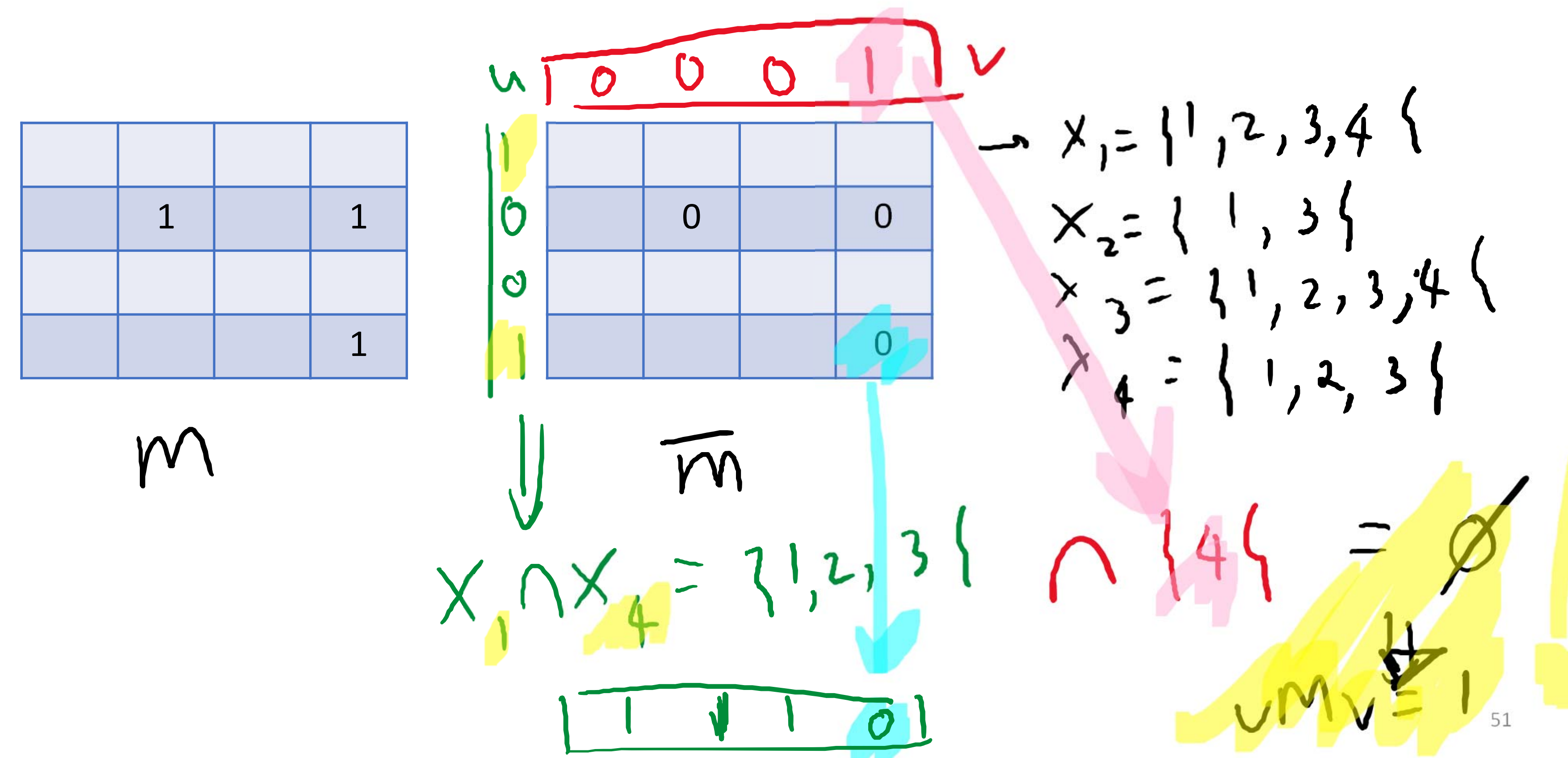
$x_3 = \{1, 2, 3, 4\}$

$x_4 = \{1, 2, 3\}$

# Pagh's problem -- Reduction



# Pagh's problem -- Reduction



# Questions?

**Thanks to co-authors:**
Sayan Bhattacharya, Jan van den Brand, Deeparnab Chakraborty, Sebastian Forster, Monika Henzinger, Christian Wulff-Nilsen, Thatchaphol Saranurak

This project has received funding from the *European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme* under grant agreement No 715672