Lecture 10: Dynamic Graph Problems

<u>Plan:</u>

➡ Introduction

➡ Fine-grained lower bounds
via "static to dynamic" reductions

➡ A new "dynamic" conjecture

Amir Abboud 3.1.22

Dynamic graph algorithms

Given initial graph G, can preprocess it.
Edge updates: insert(u,v), delete(u,v)

Queries: (depend on the problem)
How many SCCs are there? Can u reach v? …

Want to minimize the preprocessing, *update* and *query* times.

$\tilde{O}(m)$ $\tilde{O}(1)$ $\tilde{O}(1)$

- Worst case time
- Amortized time
- Total time (over all updates)

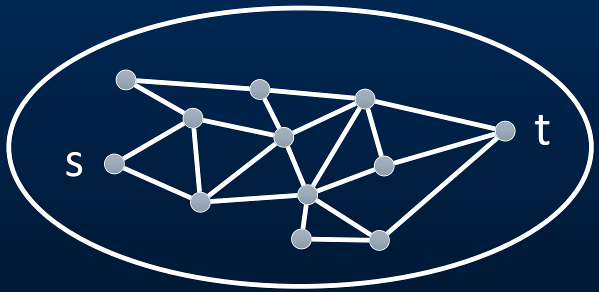
Dynamic Problems

Dynamic (undirected) Connectivity

Input: an undirected graph G

Updates: Add or remove edges.

Query: Are s and t connected?



Trivial algorithm: O(m) updates. O(1) query

[Henzinger-King '95, Thorup'01]: $O(\log m\ (\log\log m)^3)$ amortized time per update.

[Pătraşcu - Demaine STOC'05]:
$\Omega(\log m)$ Cell-probe lower bound.

Great!

Dynamic Problems

Dynamic (directed) Reachability

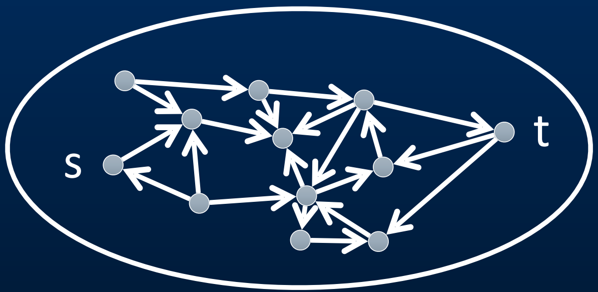
Input: A directed graph G.

Updates: Add or remove edges.

Query:

s,t-Reach: Is there a path from s to t?

#SSR: How many nodes can s reach?



Trivial algorithm: O(m) time updates

Using fast matrix multiplication
[Sankowski FOCS'04] $O(n^{1.57})$

Not great.

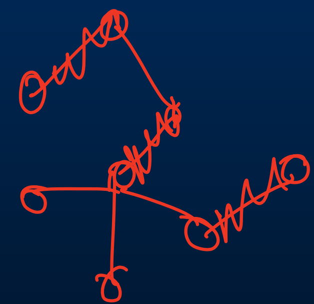
Best cell probe lower bound still $\Omega(\log m)$

Many Examples

Connectivity $\tilde{O}(1)$

Minimum Spanning Tree (MST) $\tilde{O}(1)$

Maximal Matching $\tilde{O}(1)$



Reachability

Strongly Connected Components (SCC)

s,t-shortest-path

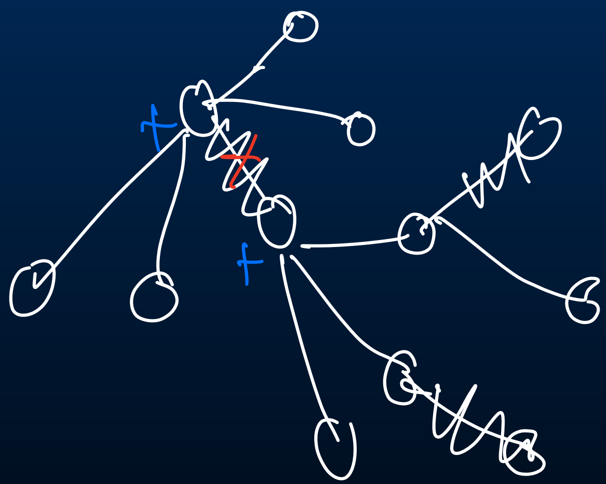
(Bipartite) Maximum Matching

s,t-Max Flow

Diameter no $\tilde{O}(1)$ update & query

Some ideas for upper bounds

Maximal Matching



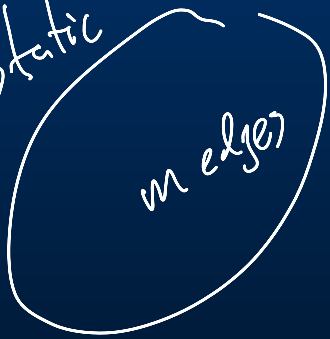
$\delta(u)$

$$O\left(\frac{deg(u)+deg(v)}{deg(u)}\right)$$

Immediate Lower Bounds

any P

Static



- add the m edges $\tilde{O}(1)$
- make 1 query $\tilde{O}(1)$

$\Downarrow$

$\tilde{O}(m)$ for P

SETH $\Rightarrow$ need $\Omega(n^{1.9})$ for Diameter even when $m = \tilde{O}(n)$.

Conditional Lower Bounds?

Connectivity

Minimum Spanning Tree (MST)

Maximal Matching

Reachability

Strongly Connected Components (SCC)

s,t-shortest-path

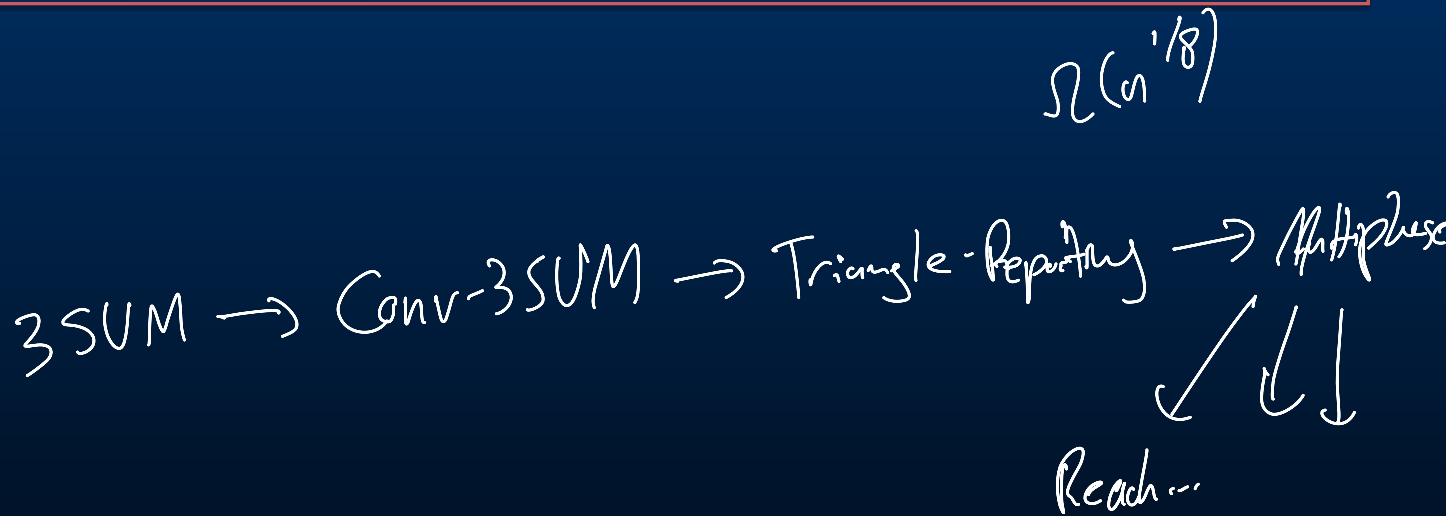
(Bipartite) Maximum Matching

s,t-Max Flow

Diameter

Conditional Lower Bounds?

[Pătraşcu STOC'10]: Polynomial Lower Bounds under the 3-SUM Conjecture.



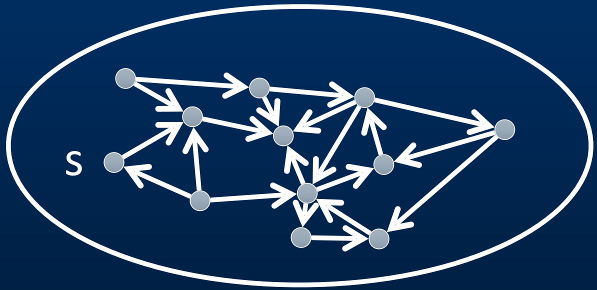
Conditional Lower Bounds?

[Pătraşcu STOC'10]: Polynomial Lower Bounds under the 3-SUM Conjecture.

[A. - Vassilevska Williams FOCS'14]: Tight lower bounds under SETH/APSP/more.

"Finding the right conjecture is the key…"

Single Source Reachability



Input: A directed graph G.

Updates: Add or remove edges.

Query:

#SSR: How many nodes can s reach?

Trivial algorithm: O(m) updates.

Theorem:

If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

$m = \tilde{O}(n)$

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof outline:

Orthogonal Vectors

(0,0,1,…,1)	(0,0,1,…,1)
(0,1,1,…,1)	(1,0,1,…,1)
…	…
(1,0,1,…,0)	(1,1,1,…,1)

Given two lists of n vectors in $\{0,1\}^d$ is there an orthogonal pair?

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof outline:

Orthogonal Vectors

(0,0,1,…,1)	(0,0,1,…,1)
(0,1,1,…,1)	(1,0,1,…,1)
…	…
(1,0,1,…,0)	(1,1,1,…,1)

(1,0,1,…,0)

(0,0,1,…,1)

Given two lists of n vectors in $\{0,1\}^d$ is there an orthogonal pair?

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof outline:

Orthogonal Vectors

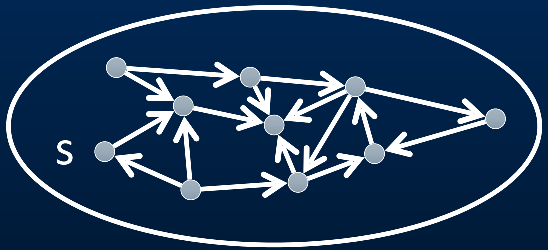
(0,0,1,…,1)	(0,0,1,…,1)
(0,1,1,…,1)	(1,0,1,…,1)
…	…
(1,0,1,…,0)	(1,1,1,…,1)

(1,0,1,…,0)

(0,0,1,…,1)

Given two lists of n vectors in $\{0,1\}^d$ is there an orthogonal pair?

dynamic #SSR



#SSR asks how many nodes can s reach?

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof outline:

Orthogonal Vectors

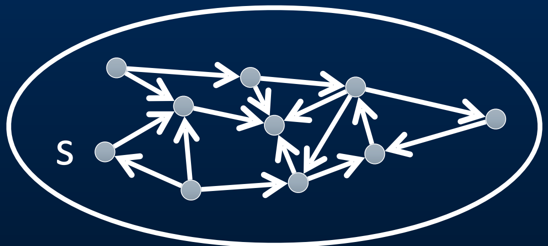
(0,0,1,…,1)	(0,0,1,…,1)
(0,1,1,…,1)	(1,0,1,…,1)
…	…
(1,0,1,…,0)	(1,1,1,…,1)

(1,0,1,…,0)

(0,0,1,…,1)

Given two lists of n vectors in $\{0,1\}^d$ is there an orthogonal pair?

dynamic #SSR



#SSR asks how many nodes can s reach?

Graph G on ***m=O(nd)*** *nodes and edges,* **O(nd)** updates and queries

d=polylog(n), m=~O(n)

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof outline:

Orthogonal Vectors

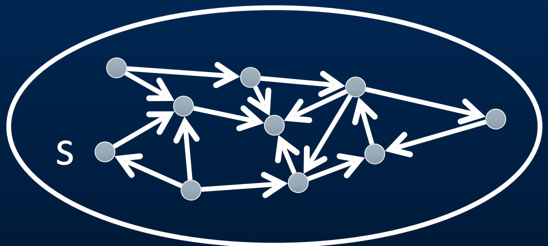
(0,0,1,…,1)	(0,0,1,…,1)
(0,1,1,…,1)	(1,0,1,…,1)
…	…
(1,0,1,…,0)	(1,1,1,…,1)

(1,0,1,…,0)

(0,0,1,…,1)

Given two lists of n vectors in $\{0,1\}^d$ is there an orthogonal pair?

dynamic #SSR



#SSR asks how many nodes can s reach?

Graph G on ***m=O(nd)*** *nodes and edges,* **O(nd)** updates and queries

Amortized ***O(m^0.9)*** update/query time

d=polylog(n), m=~O(n)

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof outline:

Orthogonal Vectors

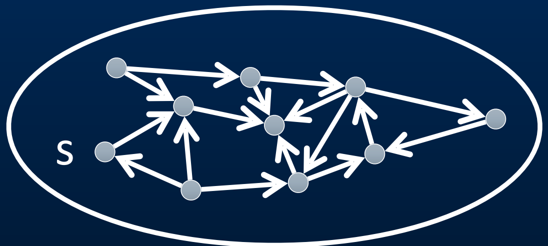
(0,0,1,…,1)	(0,0,1,…,1)
(0,1,1,…,1)	(1,0,1,…,1)
…	…
(1,0,1,…,0)	(1,1,1,…,1)

(1,0,1,…,0)

(0,0,1,…,1)

Given two lists of n vectors in $\{0,1\}^d$ is there an orthogonal pair?

dynamic #SSR



#SSR asks how many nodes can s reach?

Graph G on ***m=O(nd)*** *nodes and edges,* **O(nd)** updates and queries

O(nd) updates/queries in ***~O(n^1.9)*** time

Amortized ***O(m^0.9)*** update/query time

d=polylog(n), m=~O(n)

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof outline:

Orthogonal Vectors

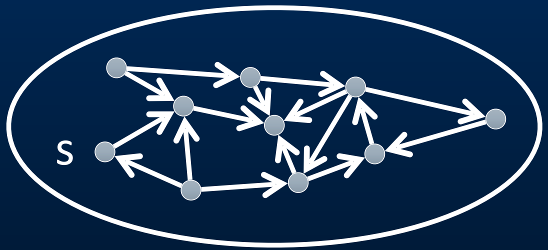
(0,0,1,…,1)	(0,0,1,…,1)
(0,1,1,…,1)	(1,0,1,…,1)
…	…
(1,0,1,…,0)	(1,1,1,…,1)

(1,0,1,…,0)

(0,0,1,…,1)

Given two lists of n vectors in $\{0,1\}^d$ is there an orthogonal pair?

dynamic #SSR



#SSR asks how many nodes can s reach?

Graph G on ***m=O(nd)*** *nodes and edges*, **O(nd)** updates and queries

OVP in ***~O(n^1.9)*** time ← ***O(nd)*** updates/queries in ***~O(n^1.9)*** time ← Amortized ***O(m^0.9)*** update/query time

(refutes SETH)

d=polylog(n), m=~O(n)

Orthogonal Vectors

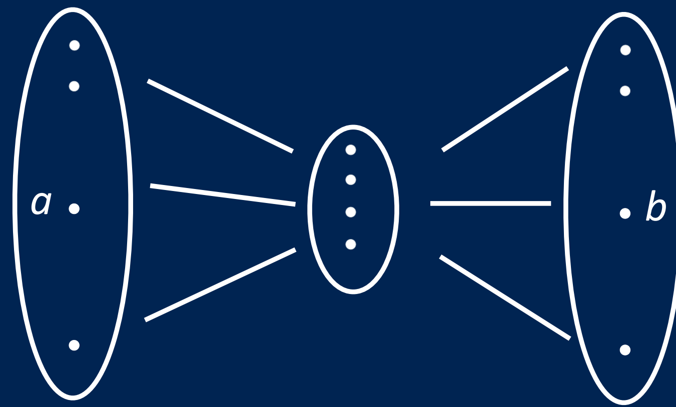
(0,0,1,…,1)	(0,0,1,…,1)
(0,1,1,…,1)	(1,0,1,…,1)
…	…
(1,0,1,…,0)	(1,1,1,…,1)

(1,0,1,…,0)

(0,0,1,…,1)

Given two lists of n vectors in $\{0,1\}^d$
is there an orthogonal pair?

Graph OV



Orthogonal Vectors

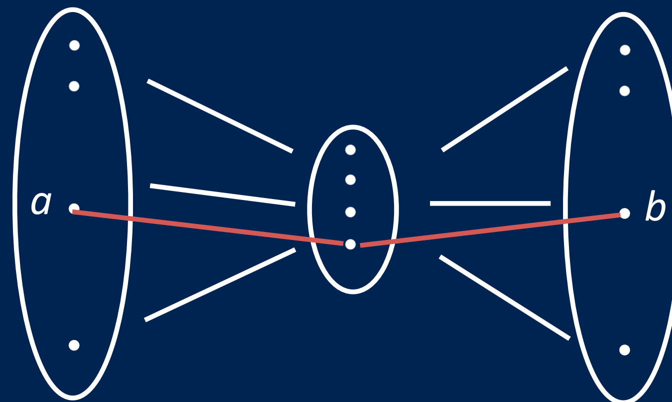
(0,0,1,…,1)	(0,0,1,…,1)
(0,1,1,…,1)	(1,0,1,…,1)
…	…
(1,0,1,…,0)	(1,1,1,…,1)

(1,0,1,…,0)

(0,0,1,…,1)

Given two lists of n vectors in $\{0,1\}^d$
is there an orthogonal pair?

Graph OV



d(a,b) = 2 if not orth.
d(a,b)>2 if orth.

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof:

Orthogonal Vectors → **dynamic #SSR**

a_1	(0,0,1,…,1)	(0,0,1,…,1)	b_1
	(0,1,1,…,1)	(1,0,1,…,1)	
	…	…	
a_n	(1,0,1,…,0)	(1,1,1,…,1)	b_n

(1,0,1,…,0)

(0,0,1,…,1)

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof: **Orthogonal Vectors** → **dynamic #SSR**

	A	B	
a_1	(0,0,1,…,1)	(0,0,1,…,1)	b_1
	(0,1,1,…,1)	(1,0,1,…,1)	
	…	…	
a_n	(1,0,1,…,0)	(1,1,1,…,1)	b_n

(1,0,1,…,0)

(0,0,1,…,1)

s •

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof:

Orthogonal Vectors → **dynamic #SSR**

a_1	(0,0,1,…,1)	(0,0,1,…,1)	b_1
	(0,1,1,…,1)	(1,0,1,…,1)	
	…	…	
a_n	(1,0,1,…,0)	(1,1,1,…,1)	b_n

(1,0,1,…,0)

(0,0,1,…,1)

s

b_1

b_n

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

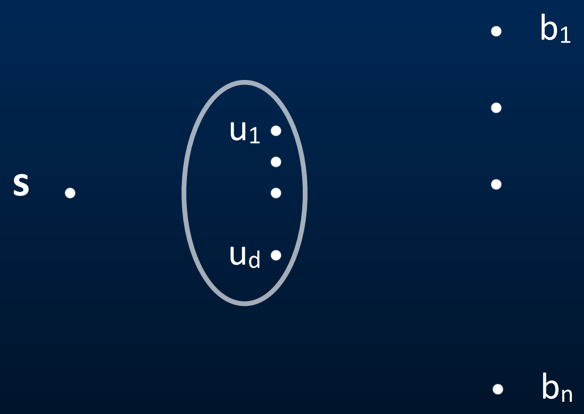
Proof:

Orthogonal Vectors → **dynamic #SSR**

a_1 (0,0,1,…,1) | (0,0,1,…,1) b_1
(0,1,1,…,1) | (1,0,1,…,1)
… | …
a_n (1,0,1,…,0) | (1,1,1,…,1) b_n

(1,0,1,…,0)

(0,0,1,…,1)



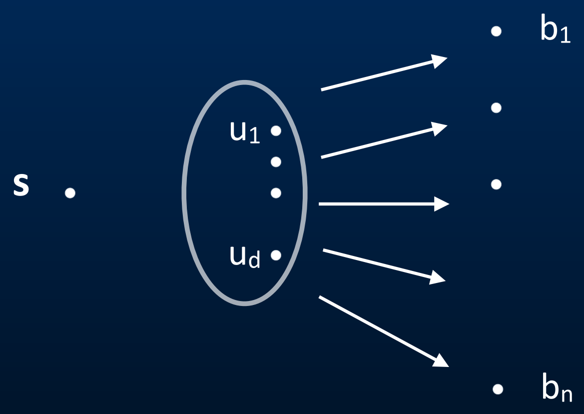
Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof: **Orthogonal Vectors** → **dynamic #SSR**

	A	B	
a_1	(0,0,1,...,1)	(0,0,1,...,1)	b_1
	(0,1,1,...,1)	(1,0,1,...,1)	
	...	...	
a_n	(1,0,1,...,0)	(1,1,1,...,1)	b_n

(1,0,1,...,0)

(0,0,1,...,1)



Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof:

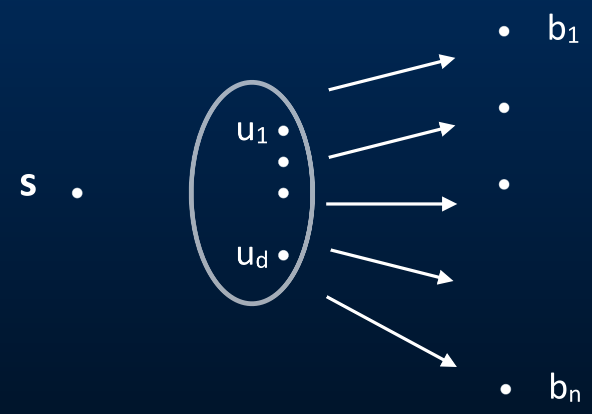
Orthogonal Vectors → **dynamic #SSR**

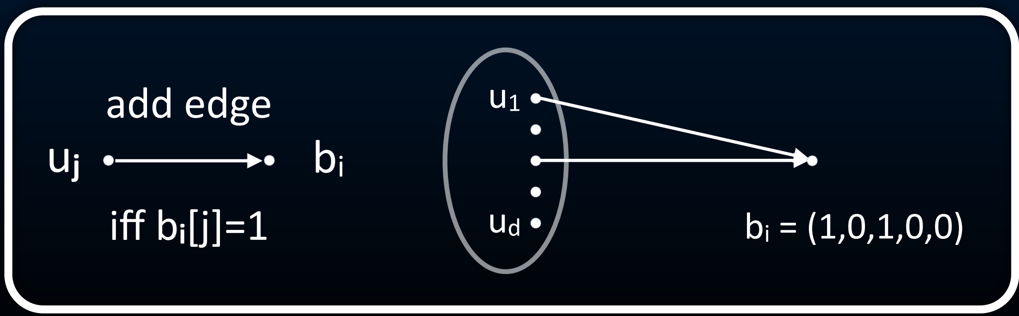
a_1 (0,0,1,…,1)
(0,1,1,…,1)
…
a_n (1,0,1,…,0)

(0,0,1,…,1) b_1
(1,0,1,…,1)
…
(1,1,1,…,1) b_n

(1,0,1,…,0)

(0,0,1,…,1)





Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof: **Orthogonal Vectors** → **dynamic #SSR**

a_1 (0,0,1,…,1) | (0,0,1,…,1) b_1

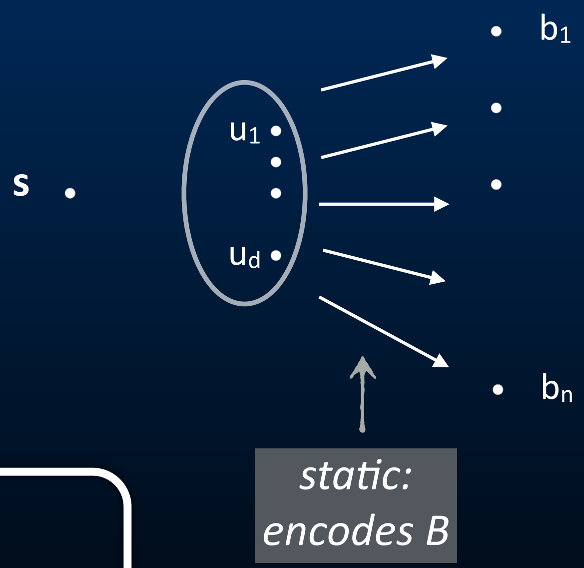
(0,1,1,…,1) | (1,0,1,…,1)

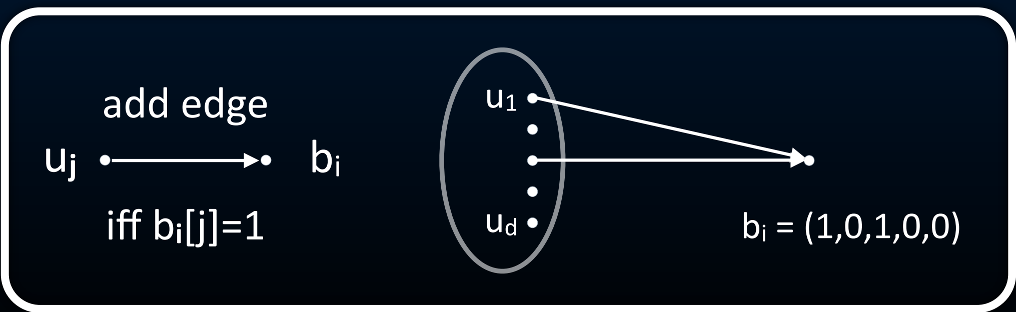
… | …

a_n (1,0,1,…,0) | (1,1,1,…,1) b_n

(1,0,1,…,0)

(0,0,1,…,1)





Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

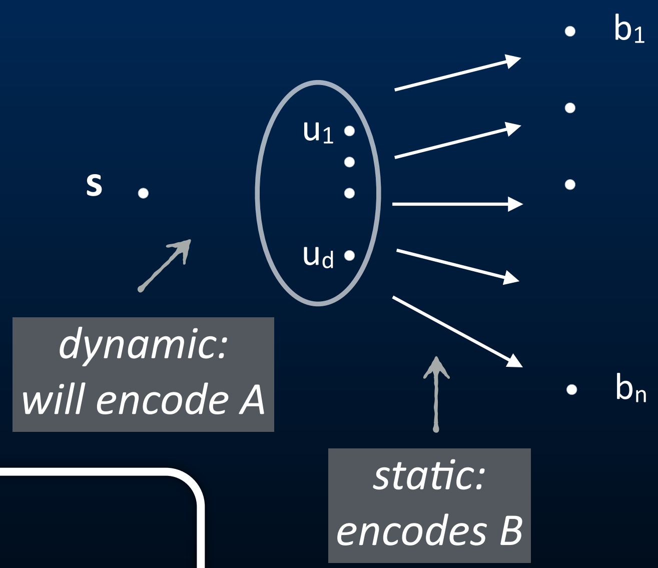
Proof:

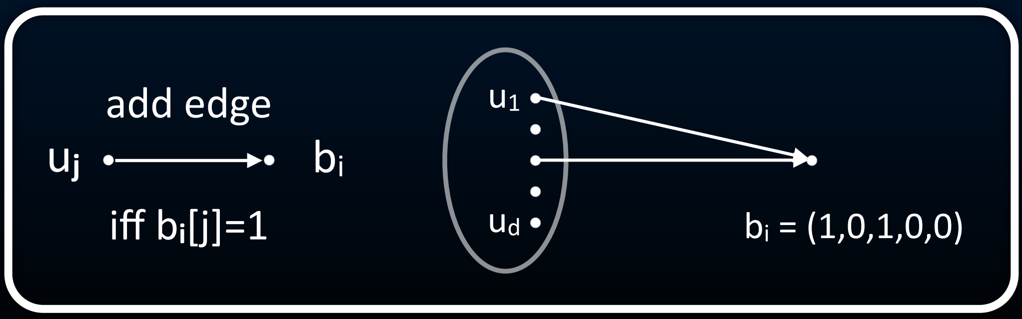
Orthogonal Vectors → **dynamic #SSR**

	A	B	
a_1	(0,0,1,…,1)	(0,0,1,…,1)	b_1
	(0,1,1,…,1)	(1,0,1,…,1)	
	…	…	
a_n	(1,0,1,…,0)	(1,1,1,…,1)	b_n

(1,0,1,…,0)

(0,0,1,…,1)





Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

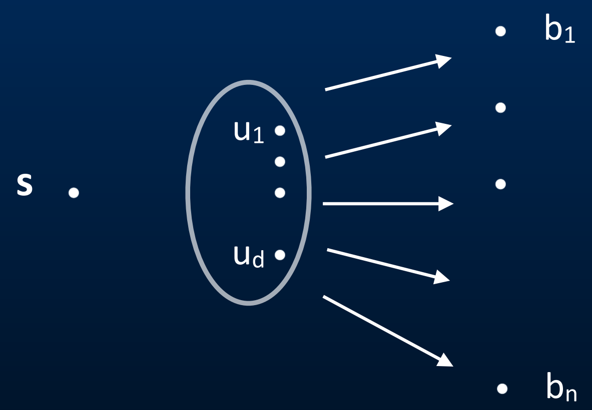
Proof:

Orthogonal Vectors → **dynamic #SSR**

a_1	(0,0,1,…,1)	(0,0,1,…,1)	b_1
	(0,1,1,…,1)	(1,0,1,…,1)	
	…	…	
a_n	(1,0,1,…,0)	(1,1,1,…,1)	b_n

(1,0,1,…,0)

(0,0,1,…,1)



add edge

$u_j \longrightarrow b_i$

iff $b_i[j]=1$

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

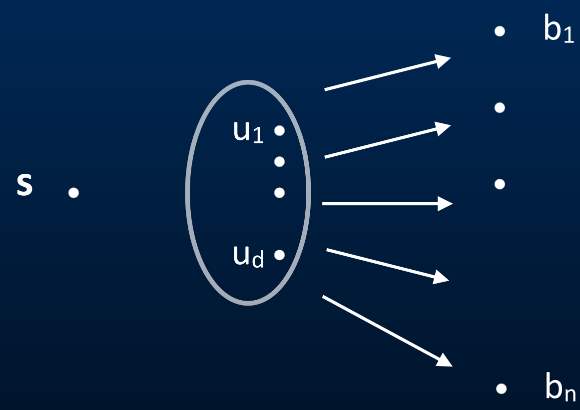
Proof:

Orthogonal Vectors → **dynamic #SSR**

	A	B	
a_1	(0,0,1,…,1)	(0,0,1,…,1)	b_1
	(0,1,1,…,1)	(1,0,1,…,1)	
	…	…	
a_n	(1,0,1,…,0)	(1,1,1,…,1)	b_n

(1,0,1,…,0)

(0,0,1,…,1)



add edge

u_j → b_i

iff $b_i[j]=1$

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

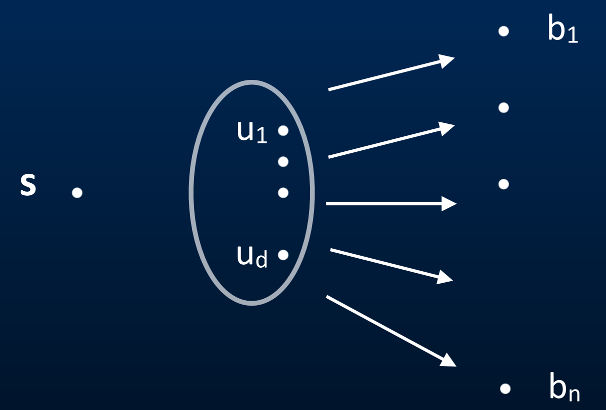
Proof:

Orthogonal Vectors → **dynamic #SSR**

	A	B	
a_1	(0,0,1,…,1)	(0,0,1,…,1)	b_1
→	(0,1,1,…,1)	(1,0,1,…,1)	
	…	…	
a_n	(1,0,1,…,0)	(1,1,1,…,1)	b_n

(1,0,1,…,0)

(0,0,1,…,1)



add edge

u_j •——→• b_i

iff $b_i[j]=1$

For each a_i:

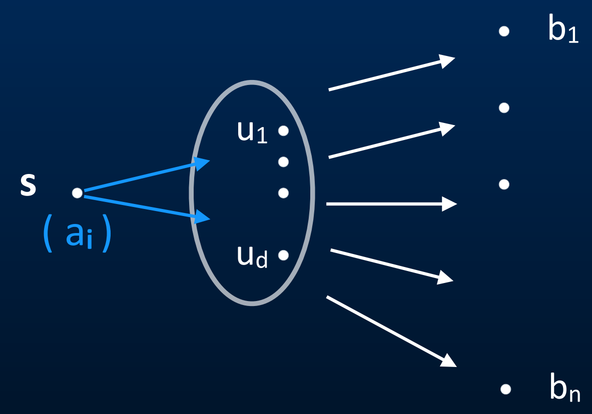
Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof: **Orthogonal Vectors** → **dynamic #SSR**

	a vectors	b vectors	
a_1	(0,0,1,…,1)	(0,0,1,…,1)	b_1
	(0,1,1,…,1)	(1,0,1,…,1)	
	…	…	
a_n	(1,0,1,…,0)	(1,1,1,…,1)	b_n

(1,0,1,…,0)

(0,0,1,…,1)



add edge u_j → b_i iff $b_i[j]=1$

For each a_i:

1. add edges s → u_j iff $a_i[j]=1$

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof: **Orthogonal Vectors** → **dynamic #SSR**

a_1 (0,0,1,…,1) | (0,0,1,…,1) b_1

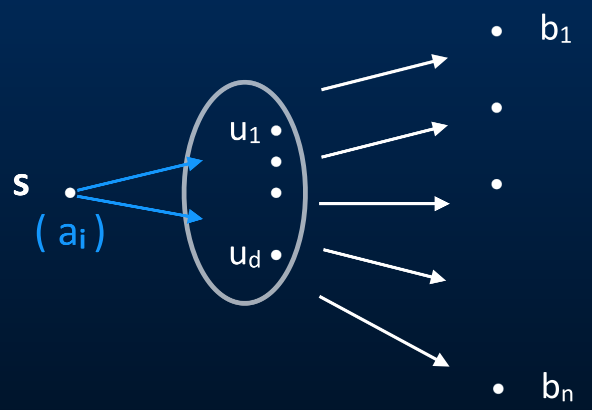
(0,1,1,…,1) | (1,0,1,…,1)

… | …

a_n (1,0,1,…,0) | (1,1,1,…,1) b_n

(1,0,1,…,0)

(0,0,1,…,1)



add edge

u_j → b_i

iff $b_i[j]=1$

For each a_i:

1. add edges $s \to u_j$ iff $a_i[j]=1$
2. ask #SSR(s)

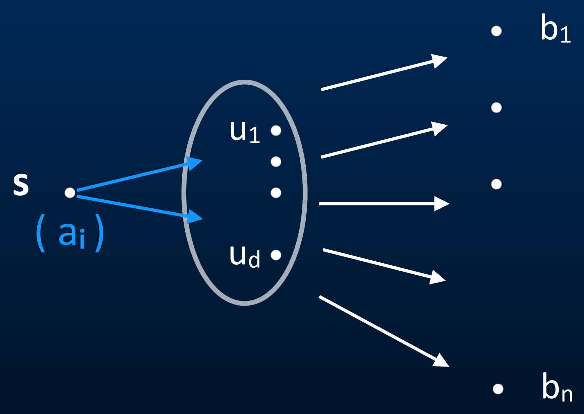
<u>Theorem</u>: If **<u>dynamic #SSR</u>** can be solved with $O(m^{0.99})$ update and query times, then **<u>OV</u>** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

<u>Proof:</u> **Orthogonal Vectors** → **dynamic #SSR**

a_1	(0,0,1,…,1)	(0,0,1,…,1)	b_1
→	(0,1,1,…,1)	(1,0,1,…,1)	
	…	…	
a_n	(1,0,1,…,0)	(1,1,1,…,1)	b_n

(1,0,1,…,0)

(0,0,1,…,1)



add edge

u_j → b_i

iff $b_i[j]=1$

<u>Observation</u>:

s cannot reach ***b*** iff $\boldsymbol{a_i}$ and ***b*** are orthogonal.

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

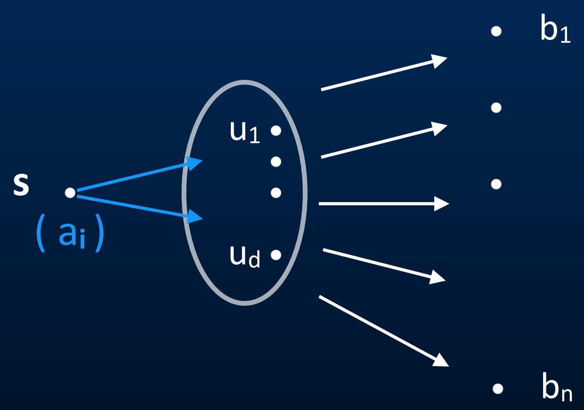
Proof: **Orthogonal Vectors** → **dynamic #SSR**

a_1 (0,0,1,…,1)
(0,1,1,…,1)
…
a_n (1,0,1,…,0)

b_1 (0,0,1,…,1)
(1,0,1,…,1)
…
b_n (1,1,1,…,1)

(1,0,1,…,0)

(0,0,1,…,1)



add edge $u_j \to b_i$ iff $b_i[j]=1$

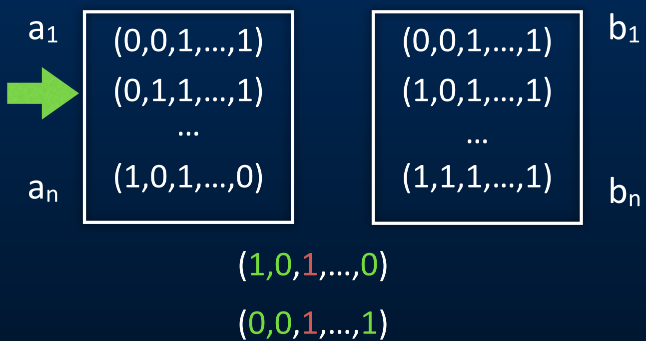
For each a_i:

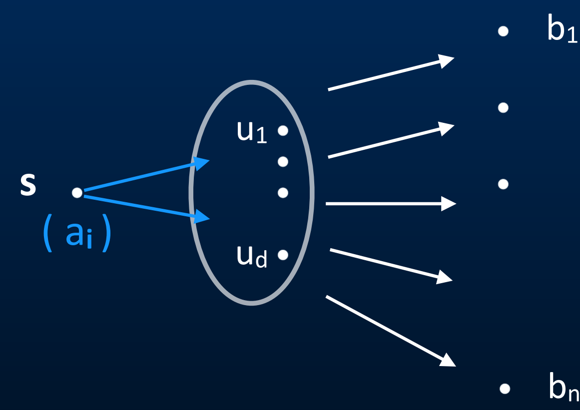
1. add edges $s \to u_j$ iff $a_i[j]=1$
2. ask #SSR(s),

   if *< n + (1s in a_i)*, output "yes".

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof: **Orthogonal Vectors** → **dynamic #SSR**





add edge
u_j •——→• b_i
iff $b_i[j]=1$

For each a_i:

1. add edges s •——→• u_j iff $a_i[j]=1$
2. ask #SSR(s),
   if *< n + (1s in a_i)*, output "yes".
3. remove edges and move on to next a_i

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

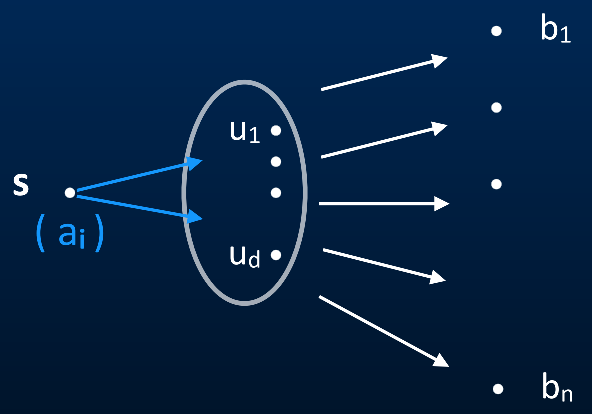
Proof:

Orthogonal Vectors → **dynamic #SSR**

a_1	(0,0,1,...,1)	(0,0,1,...,1)	b_1
	(0,1,1,...,1)	(1,0,1,...,1)	
	...	...	
a_n	(1,0,1,...,0)	(1,1,1,...,1)	b_n

(1,0,1,...,0)

(0,0,1,...,1)



O(nd) updates,
m = O(nd) edges

For each a_i:

1. add edges s → u_j iff $a_i[j]=1$
2. ask #SSR(s),
   and if ***< n + (1s in a_i)***, output "yes".
3. remove edges and move on to next a_i

Theorem: If **dynamic #SSR** can be solved with $O(m^{0.99})$ update and query times, then **OV** can be solved in $\tilde{O}(n^{1.99})$ time (and SETH is false).

Proof:

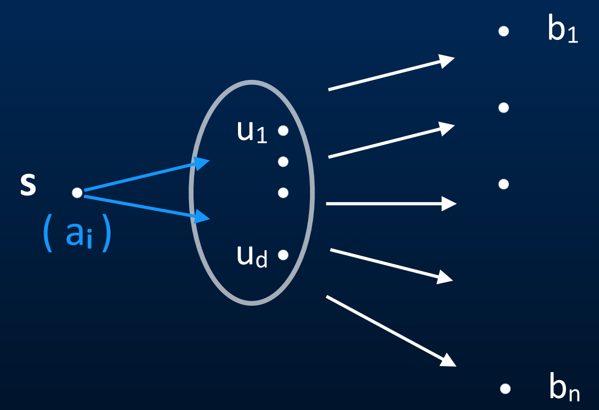
Orthogonal Vectors → **dynamic #SSR**

a_1
(0,0,1,…,1)
(0,1,1,…,1)
…
a_n
(1,0,1,…,0)

b_1
(0,0,1,…,1)
(1,0,1,…,1)
…
b_n
(1,1,1,…,1)

(1,0,1,…,0)

(0,0,1,…,1)



O(nd) updates,
m = O(nd) edges

~Ω(m) per update!

For each a_i:

1. add edges s → u_j iff $a_i[j]=1$
2. ask #SSR(s),
   and if ***< n + (1s in a_i)***, output "yes".
3. remove edges and move on to next a_i

With additional gadgets, lower bounds for:

Strongly Connected Components

Undirected Connectivity with node updates

and more.

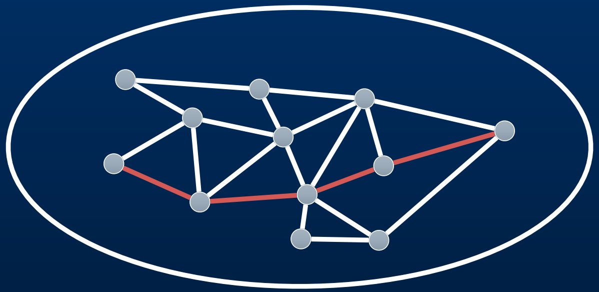
Next: even higher lower bounds!

Dynamic Diameter

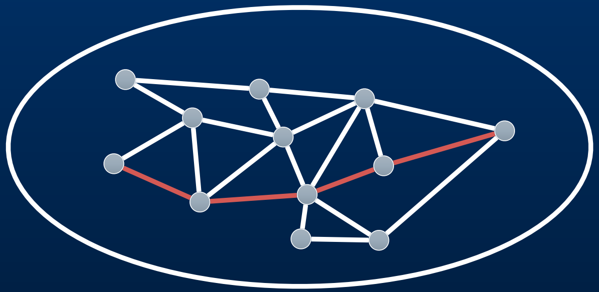
Input: an undirected graph G

Updates: Add or remove edges.

Query: What is the diameter of G?



Dynamic Diameter



Input: an undirected graph G

Updates: Add or remove edges.

Query: What is the diameter of G?

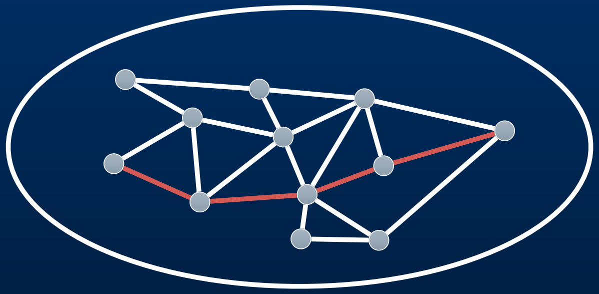
Upper bounds for dynamic All-Pairs-Shortest-Paths:
Naive: ~$O(mn)$ per update.
[Demetrescu-Italiano 03', Thorup 04']: amortized ~$O(n^2)$.

Dynamic Diameter

Input: an undirected graph G

Updates: Add or remove edges.

Query: What is the diameter of G?



Upper bounds for dynamic All-Pairs-Shortest-Paths:
Naive: ~$O(mn)$ per update.
[Demetrescu-Italiano 03', Thorup 04']: amortized ~$O(n^2)$.

Theorem: 1.3-approximation for the diameter of a sparse graph under edge updates with amortized $O(m^{1.99})$ updates refutes SETH!

Theorem: 1.3-approximation for the diameter of a sparse graph under edge updates with amortized $O(m^{1.99})$ updates refutes SETH!

Proof outline:

Three Orthogonal Vectors

(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

Given three lists of n vectors in $\{0,1\}^d$
is there an "orthogonal" triple?

Theorem: 1.3-approximation for the diameter of a sparse graph under edge updates with amortized $O(m^{1.99})$ updates refutes SETH!

Proof outline:

Three Orthogonal Vectors

(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)

Given three lists of n vectors in $\{0,1\}^d$ is there an "orthogonal" triple?

Theorem: 1.3-approximation for the diameter of a sparse graph under edge updates with amortized $O(m^{1.99})$ updates refutes SETH!

Proof outline:

Three Orthogonal Vectors

(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

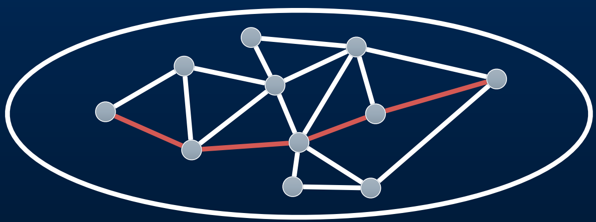
(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)

Given three lists of n vectors in $\{0,1\}^d$ is there an "orthogonal" triple?

dynamic Diameter



Theorem: 1.3-approximation for the diameter of a sparse graph under edge updates with amortized $O(m^{1.99})$ updates refutes SETH!

Proof outline:

Three Orthogonal Vectors

(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

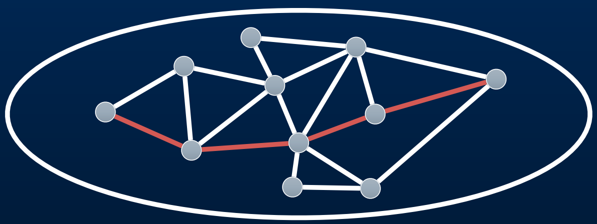
(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)

Given three lists of n vectors in $\{0,1\}^d$
is there an "orthogonal" triple?

dynamic Diameter



d=polylog(n)

Lemma: **3-OV** in $\tilde{O}(n^{3-e})$ time refutes **SETH**

Theorem: 1.3-approximation for the diameter of a sparse graph under edge updates with amortized $O(m^{1.99})$ updates refutes SETH!

Proof outline:

Three Orthogonal Vectors

(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

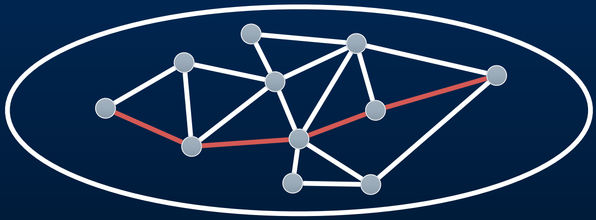
(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)

Given three lists of n vectors in $\{0,1\}^d$ is there an "orthogonal" triple?

dynamic Diameter



is the diameter 3 or more?

Graph G on ***m=O(nd)*** *nodes and edges,* **O(nd)** updates and queries

d=polylog(n), m=~O(n)

Theorem: 1.3-approximation for the diameter of a sparse graph under edge updates with amortized $O(m^{1.99})$ updates refutes SETH!

Proof outline:

Three Orthogonal Vectors

(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

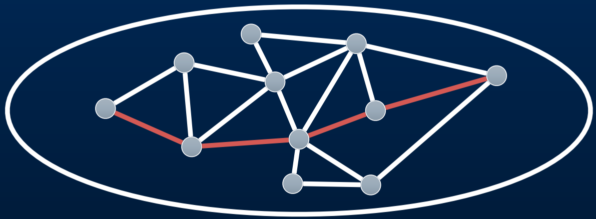
(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)

Given three lists of n vectors in $\{0,1\}^d$ is there an "orthogonal" triple?

dynamic Diameter



is the diameter 3 or more?

Graph G on ***m=O(nd)*** *nodes and edges*, **O(nd)** updates and queries

Amortized ***$O(m^{1.9})$*** update/query time

d=polylog(n), m=~O(n)

Theorem: 1.3-approximation for the diameter of a sparse graph under edge updates with amortized $O(m^{1.99})$ updates refutes SETH!

Proof outline:

Three Orthogonal Vectors

(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

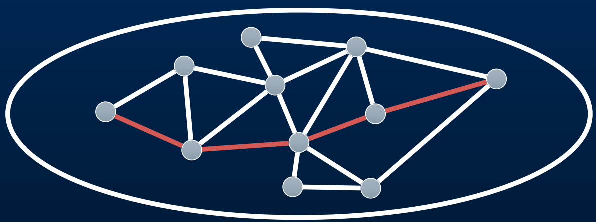
(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)

Given three lists of n vectors in $\{0,1\}^d$ is there an "orthogonal" triple?

dynamic Diameter



is the diameter 3 or more?

Graph G on ***m=O(nd)*** *nodes and edges,* **O(nd)** updates and queries

O(nd) updates/queries in ***~O(n^{2.9})*** time

Amortized ***$O(m^{1.9})$*** update/query time

d=polylog(n), m=~O(n)

Theorem: 1.3-approximation for the diameter of a sparse graph under edge updates with amortized $O(m^{1.99})$ updates refutes SETH!

Proof outline:

Three Orthogonal Vectors

(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

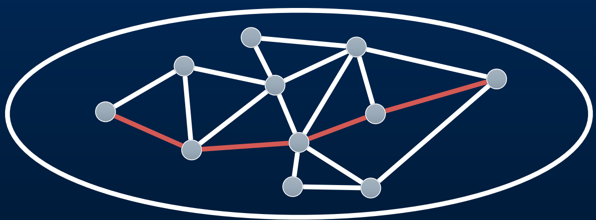
(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)

Given three lists of n vectors in $\{0,1\}^d$ is there an "orthogonal" triple?

dynamic Diameter



is the diameter 3 or more?

Graph G on ***m=O(nd)*** *nodes and edges,* **O(nd)** updates and queries

3-OVP in ***~O(n^2.9)*** time ← ***O(nd)*** updates/queries in ***~O(n^2.9)*** time ← Amortized ***O(m^1.9)*** update/query time

(refutes SETH)

d=polylog(n), m=~O(n)

Theorem: 1.3-approximation for the diameter of a sparse graph under edge updates with amortized $O(m^{1.99})$ updates refutes SETH!

Proof:

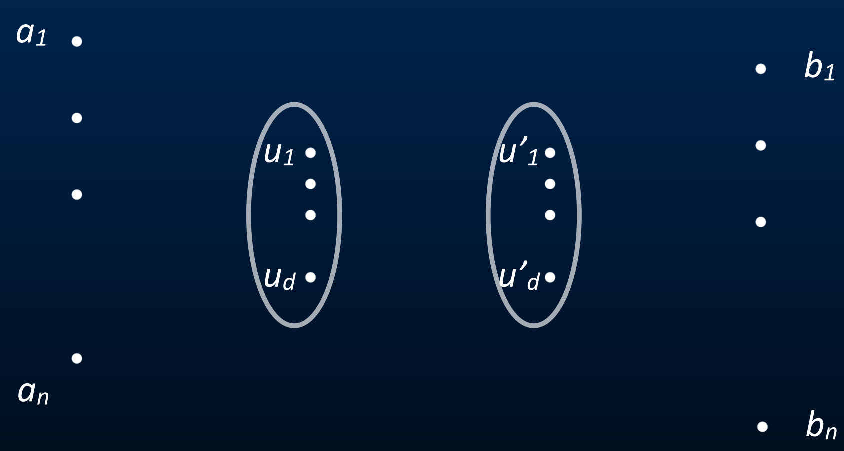
Three Orthogonal Vectors → dynamic Diameter

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)



Theorem: 1.3-approximation for the diameter of a sparse graph under edge updates with amortized $O(m^{1.99})$ updates refutes SETH!

Proof:

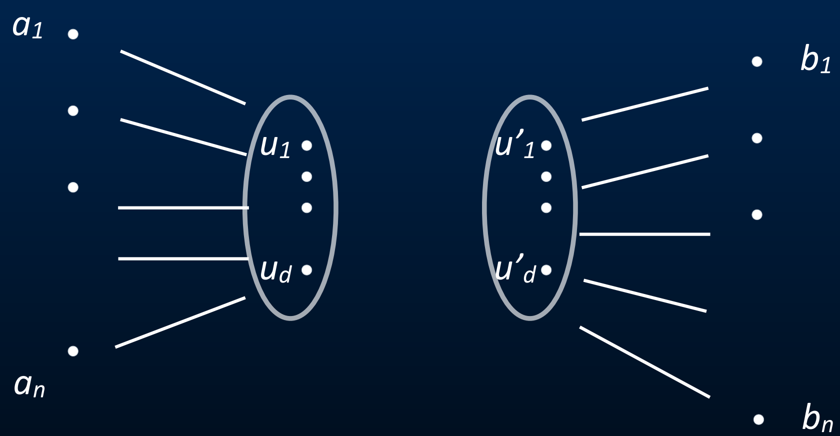
Three Orthogonal Vectors → dynamic Diameter

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)



Theorem: 1.3-approximation for the diameter of a sparse graph under edge updates with amortized $O(m^{1.99})$ updates refutes SETH!

Proof:

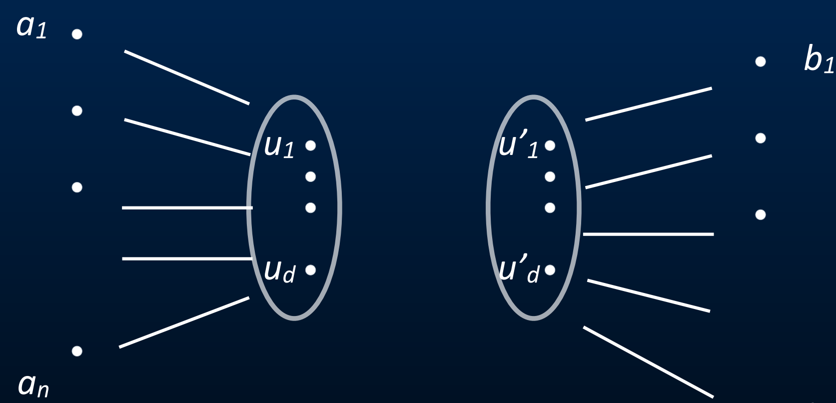
Three Orthogonal Vectors → **dynamic Diameter**

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)



add edge

u'_j •——• b_i

iff $b_i[j]=1$

Theorem: 1.3-approximation for the diameter of a sparse graph under edge updates with amortized $O(m^{1.99})$ updates refutes SETH!

Proof:

Three Orthogonal Vectors → **dynamic Diameter**

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

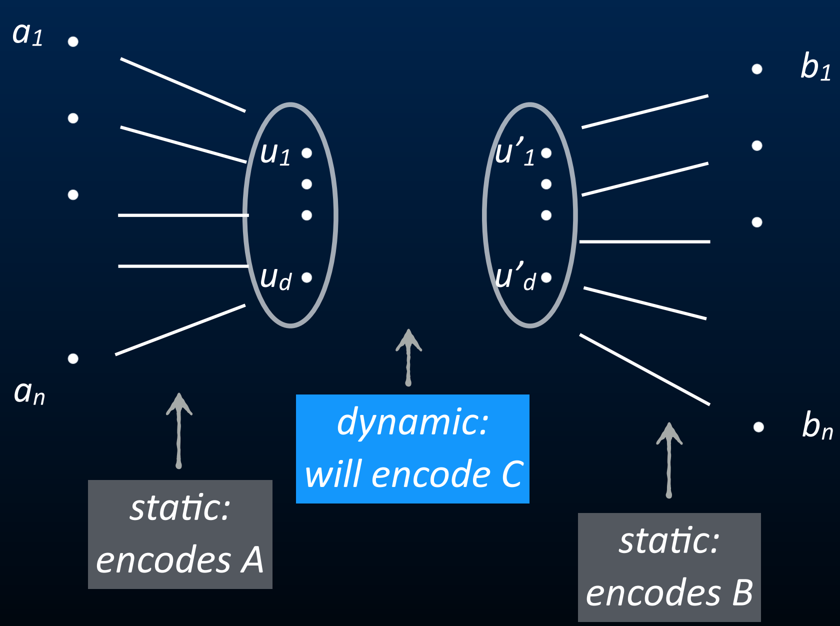
(0,1,1,…,0)

(1,1,1,…,0)

add edge

u'_j •——• b_i

iff $b_i[j]=1$



Proof:

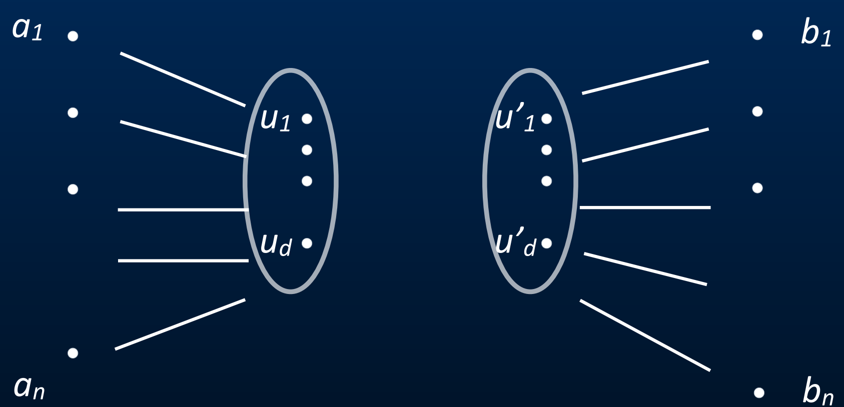
Three Orthogonal Vectors → **dynamic Diameter**

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)



add edge

u'_j •———• b_i

iff $b_i[j]=1$

Proof:

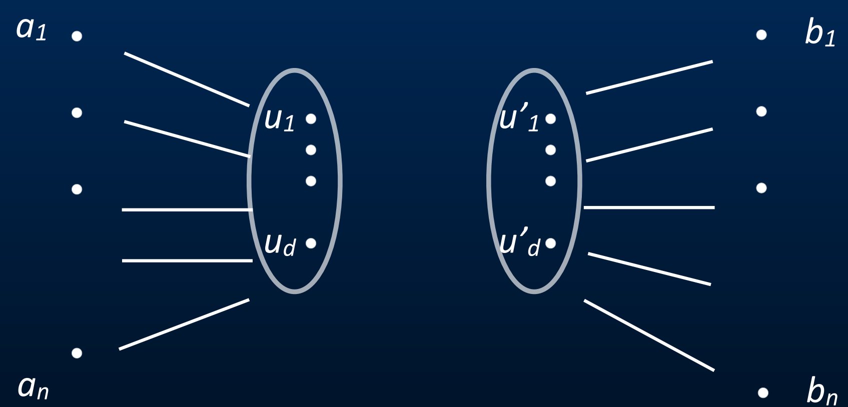
Three Orthogonal Vectors → dynamic Diameter

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)



add edge

u'_j —— b_i

iff $b_i[j]=1$

Proof:

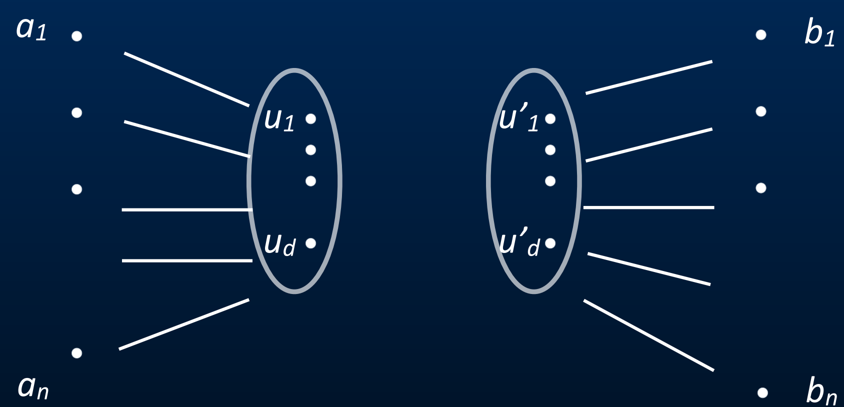
Three Orthogonal Vectors → dynamic Diameter

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)



add edge

u'_j • —— • b_i

iff $b_i[j]=1$

For each c_i:

Proof:

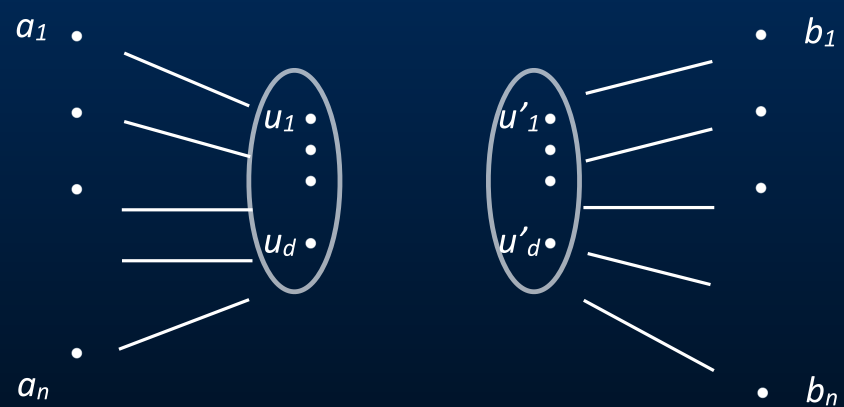
Three Orthogonal Vectors → dynamic Diameter

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)



add edge

u'_j —— b_i

iff $b_i[j]=1$

For each c_i:

1. add edges u_j —— u'_j iff $c_i[j]=1$

Proof:

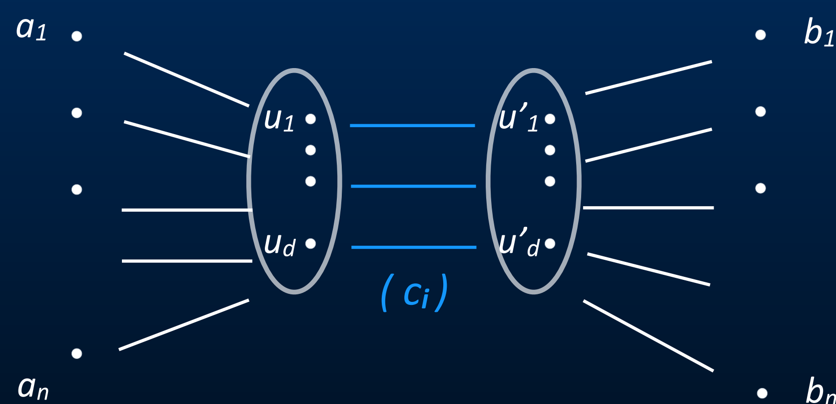
Three Orthogonal Vectors → dynamic Diameter

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)



add edge

u'_j •——• b_i

iff $b_i[j]=1$

For each c_i:

1. add edges u_j •——• u'_j iff $c_i[j]=1$

Proof:

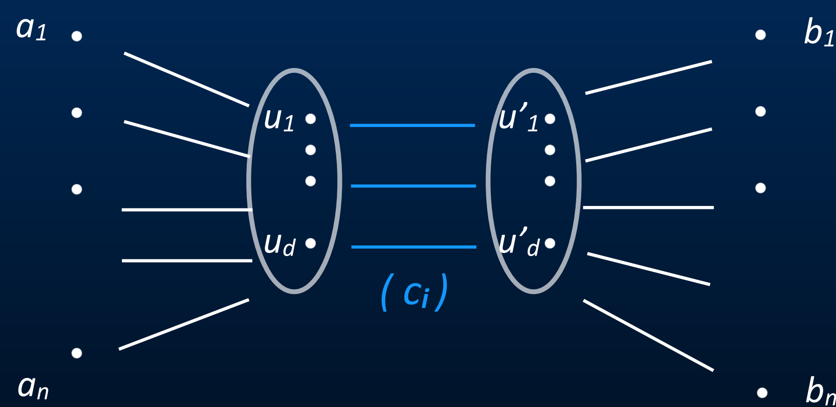
Three Orthogonal Vectors → dynamic Diameter

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)



add edge

u'_j •——• b_i

iff $b_i[j]=1$

For each c_i:

1. add edges u_j •——• u'_j iff $c_i[j]=1$
2. ask Diameter query.

Theorem: 1.3-approximation for the diameter of a sparse graph under edge updates with amortized $O(m^{1.99})$ updates refutes SETH!

Proof:

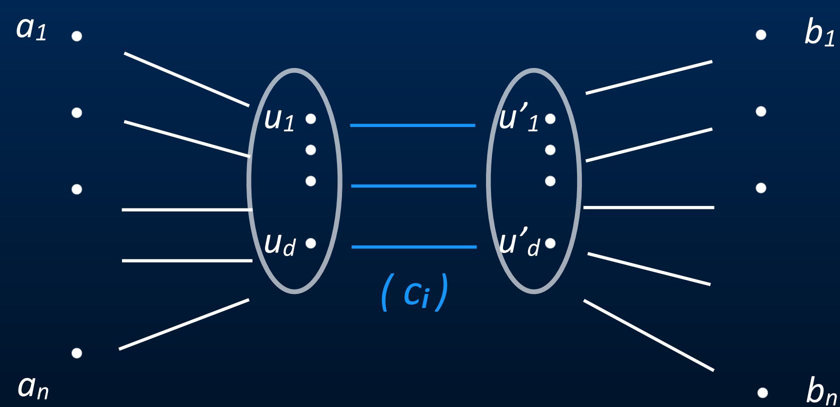
Three Orthogonal Vectors → dynamic Diameter

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)



add edge

u'_j •——• b_i

iff $b_i[j]=1$

For each c_i:

1. add edges u_j •——• u'_j iff $c_i[j]=1$
2. ask Diameter query.

Proof:

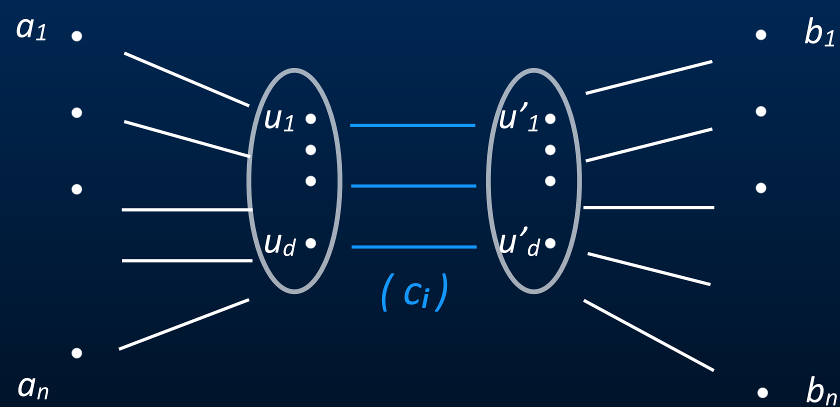
Three Orthogonal Vectors → dynamic Diameter

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)



add edge

u'_j •———• b_i

iff $b_i[j]=1$

Observation:

The distance from *a* to *b* is more than ***3*** iff a,b,c_i are an orthogonal triple.

Proof:

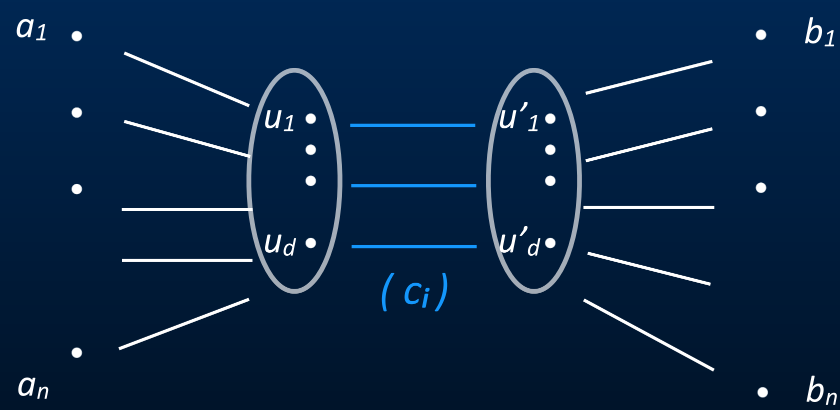
Three Orthogonal Vectors → dynamic Diameter

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)



add edge

u'_j — b_i

iff $b_i[j]=1$

Observation:

The distance from *a* to *b* is more than ***3*** iff *a,b,c_i* are an orthogonal triple.

(no coordinate with all three 1's)

Theorem: 1.3-approximation for the diameter of a sparse graph under edge updates with amortized $O(m^{1.99})$ updates refutes SETH!

Proof:

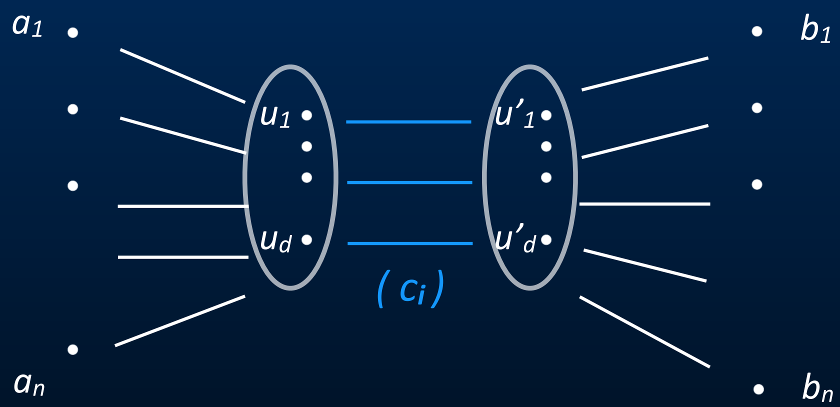
Three Orthogonal Vectors → **dynamic Diameter**

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)



add edge

u'_j •——• b_i

iff $b_i[j]=1$

Observation:
The distance from *a* to *b* is more than ***3*** iff a,b,c_i are an orthogonal triple.

(no coordinate with all three 1's)

Proof:

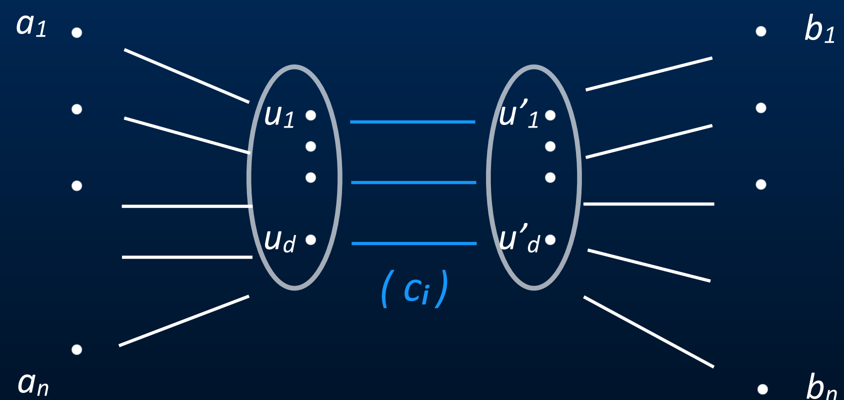
Three Orthogonal Vectors → dynamic Diameter

A
(0,0,…,1)
(0,1,…,1)
…
(1,0,…,0)

B
(1,0,…,1)
(0,1,…,0)
…
(1,0,…,1)

C
(1,0,…,1)
(0,0,…,1)
…
(1,1,…,0)

(1,0,1,…,0)
(0,1,1,…,0)
(1,1,1,…,0)



For each c_i:

1. add edges u_j ——— u'_j iff $c_i[j]=1$
2. Query. If Diameter > 3, output “yes”.

Proof:

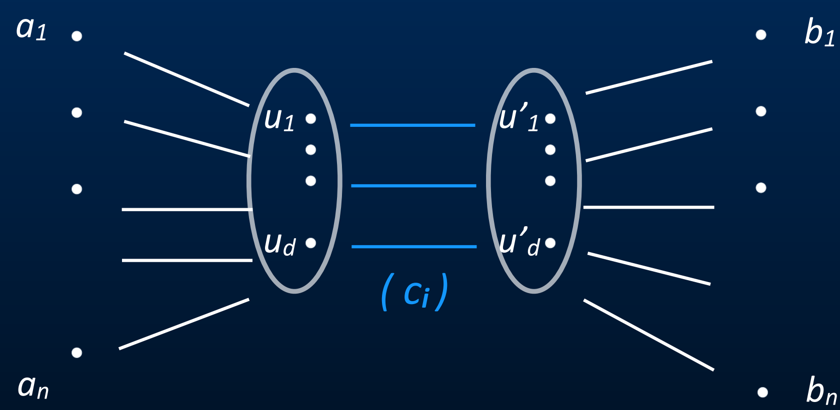
Three Orthogonal Vectors → dynamic Diameter

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)



For each c_i:

1. add edges u_j —— u'_j iff $c_i[j]=1$
2. Query. If Diameter > 3, output "yes".
3. remove edges and move on to next c_i

Proof:

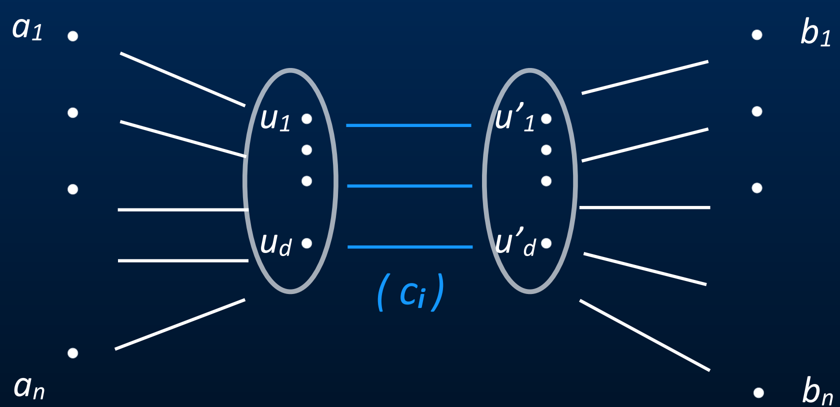
Three Orthogonal Vectors → dynamic Diameter

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)



O(nd) updates,
m = O(nd) edges

~Ω(n^2) per update!

For each c_i:

1. add edges u_j —— u'_j iff $c_i[j]=1$
2. Query. If Diameter > 3, output "yes".
3. remove edges and move on to next c_i

Theorem: 1.3-approximation for the diameter of a sparse graph under edge updates with amortized $O(m^{1.99})$ updates refutes SETH!

Proof:

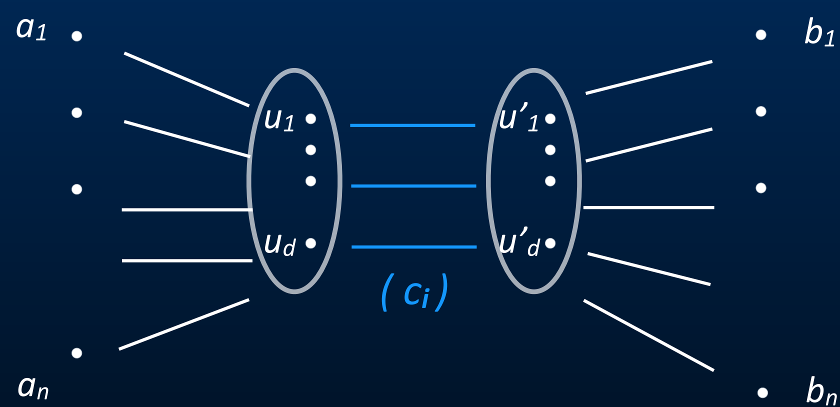
Three Orthogonal Vectors → dynamic Diameter

A	B	C
(0,0,…,1)	(1,0,…,1)	(1,0,…,1)
(0,1,…,1)	(0,1,…,0)	(0,0,…,1)
…	…	…
(1,0,…,0)	(1,0,…,1)	(1,1,…,0)

(1,0,1,…,0)

(0,1,1,…,0)

(1,1,1,…,0)



O(nd) updates,
m = O(nd) edges

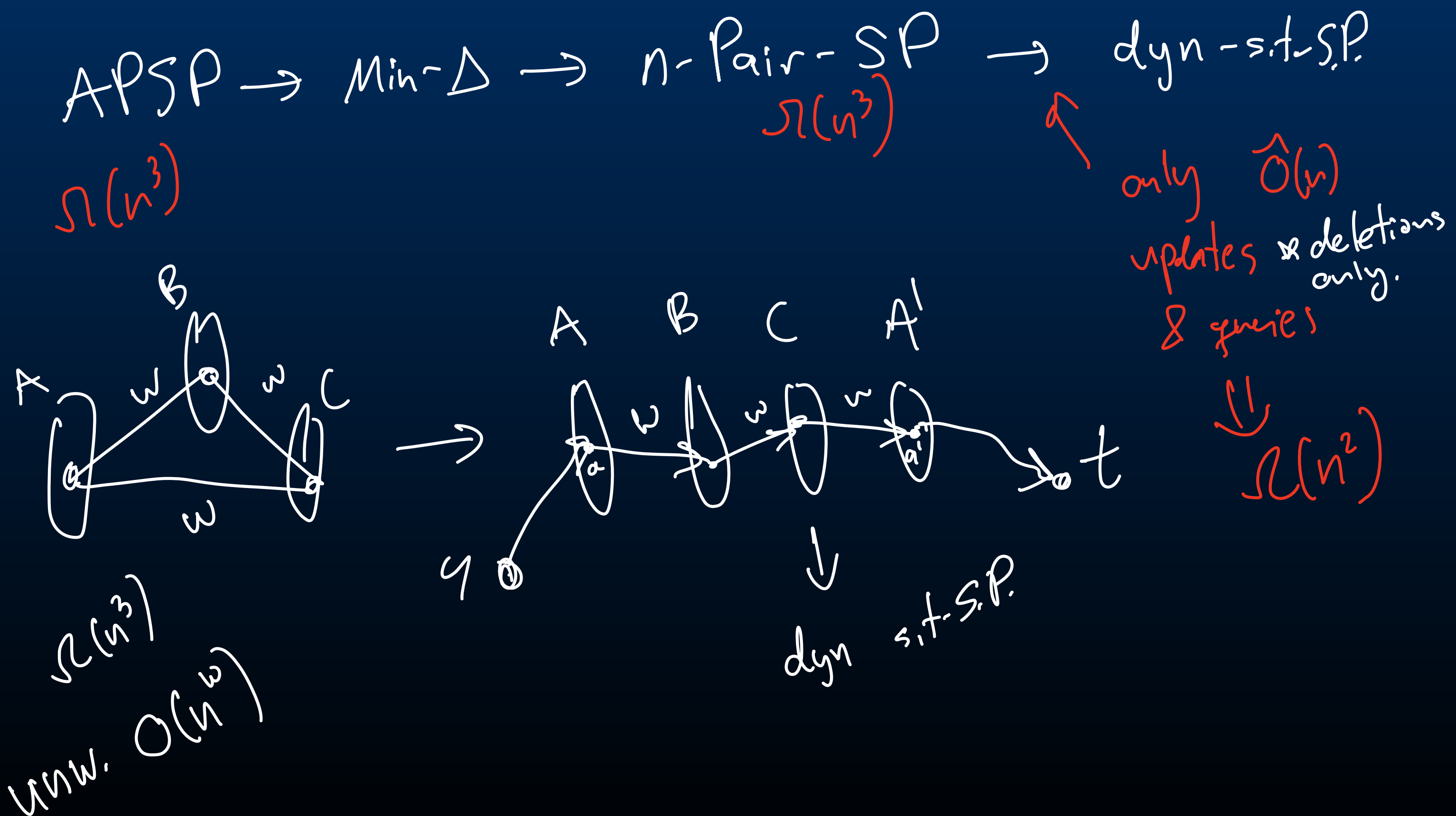
~Ω(n²) per update!

For each c_i:

1. add edges u_j —— u'_j iff $c_i[j]=1$
2. Query. If Diameter > 3, output “yes”.
3. remove edges and move on to next c_i

Shortest Path

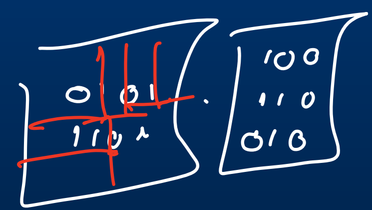
Theorem: s,t-shortest path with amortized $O(n^{1.99})$ updates refutes APSP.



What about unweighted?

"Combinatorial BMM Conj"

no $n^{2.99}$ for BMM.



Best Comb. algs:

- "Four Russians alg" $\frac{n^3}{\lg^2 n}$
- $\frac{n^3}{\lg^4 n}$

$\Omega(n^{3-\varepsilon})$ for Δ-det, combinatorially.

APSP → Min-Δ

RNA folding

$\Omega(n^3)$ - comb.

$O(n^{2.8})$ - using fMM

OMv Lower Bounds

[Henzinger - Krinninger - Nanongkai - Saranurak STOC '15]
Most BMM lower bounds hold for non-combinatorial algorithms as well, under the Online Matrix Vector Multiplication Conjecture.

OMv problem: Given n x n Boolean matrix A and n Boolean vectors $v_1,\ldots,v_n$, given online, return each $A \cdot v_i$ right after v_i has been given.

Conj: no $O(n^{3-\varepsilon})$ alg.

Gt-Williams: $\frac{n^3}{2^{\sqrt{\log n}}}$.

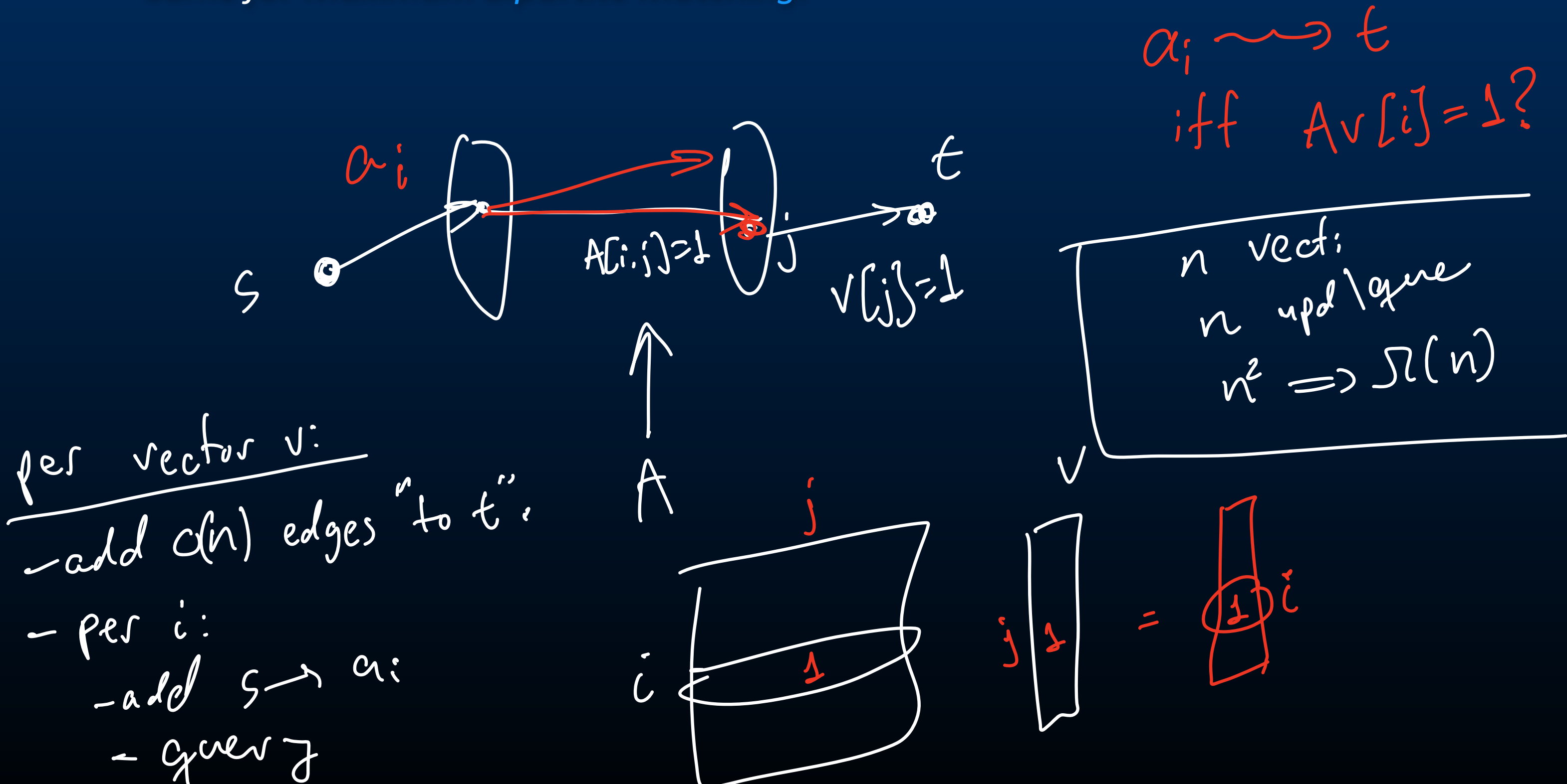
$\forall i$: $A \cdot v_i$ before seeing v_{i+1}.

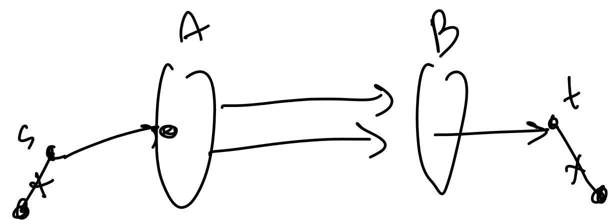
$\Omega(n^3)$.

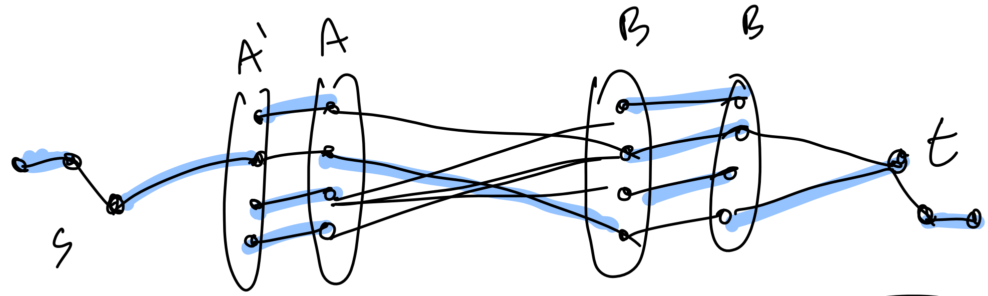
Maximum Matching

Theorem: s,t-reachability with amortized $O(n^{0.99})$ updates refutes OMv.

Same for Maximum Bipartite Matching.







(1+ε)-apx dyn. matching

$\hat{O}(1)$ updates

?