Lower Bounds for Shortest Paths and Matchings

Krzysztof Onak
CMU

Joint work with **Venkat Guruswami**

This Talk

- Quick Reminder: this is a streaming workshop

This Talk

- Quick Reminder: this is a streaming workshop
- Model:
  - Input: stream of graph edges

Algorithm $\longleftarrow \mathbf{(5,4)\ (1,2)\ (4,3)\ (2,5)\ (3,1)}\ \ldots$

This Talk

- Quick Reminder: this is a streaming workshop
- Model:
  - Input: stream of graph edges
  - Worst-case ordering (as opposed to random)

Algorithm $\longleftarrow \mathbf{(5,4)\ (1,2)\ (4,3)\ (2,5)\ (3,1)\ \ldots}$

This Talk

- Quick Reminder: this is a streaming workshop
- Model:
  - Input: stream of graph edges
  - Worst-case ordering (as opposed to random)
  - Multiple passes allowed

Algorithm $\longleftarrow$ $\mathbf{(5,4)\ (1,2)\ (4,3)\ (2,5)\ (3,1)\ \ldots}$

This Talk

- Quick Reminder: this is a streaming workshop
- Model:
  - Input: stream of graph edges
  - Worst-case ordering (as opposed to random)
  - Multiple passes allowed

Algorithm $\leftarrow$ $\mathbf{(5,4)\ (1,2)\ (4,3)\ (2,5)\ (3,1)\ \ldots}$

- Goal/main message:

Solving classic graph problems requires $n^{1+\Omega(1)}$ space with $O(1)$ passes

This Talk

- Quick Reminder: this is a streaming workshop
- Model:
  - Input: stream of graph edges
  - Worst-case ordering (as opposed to random)
  - Multiple passes allowed

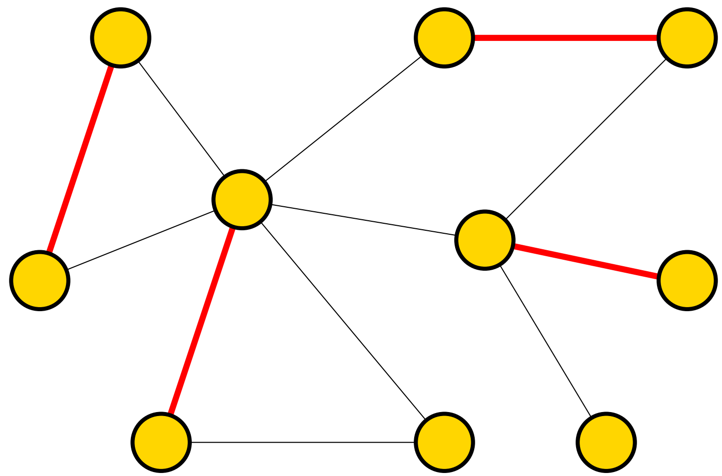
Algorithm $\longleftarrow \mathbf{(5,4)\ (1,2)\ (4,3)\ (2,5)\ (3,1)\ \ldots}$

- Goal/main message:

  Solving classic graph problems requires $\sim n^{1+\Omega(1/p)}$ space with $O(p)$ passes

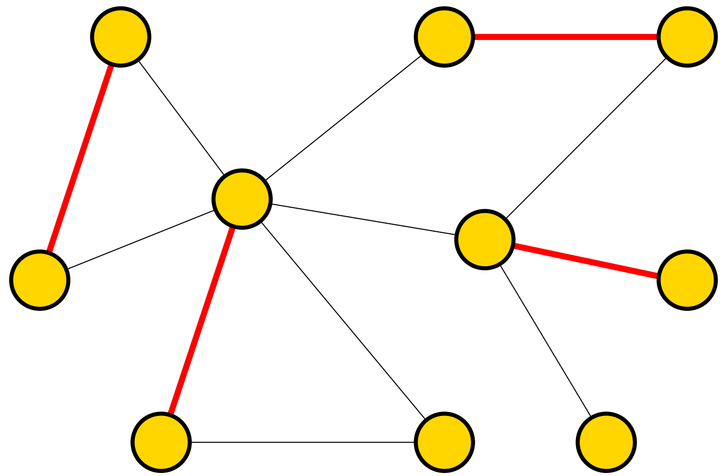
Matchings

The Maximum Matching Problem:



Matchings

The Maximum Matching Problem:

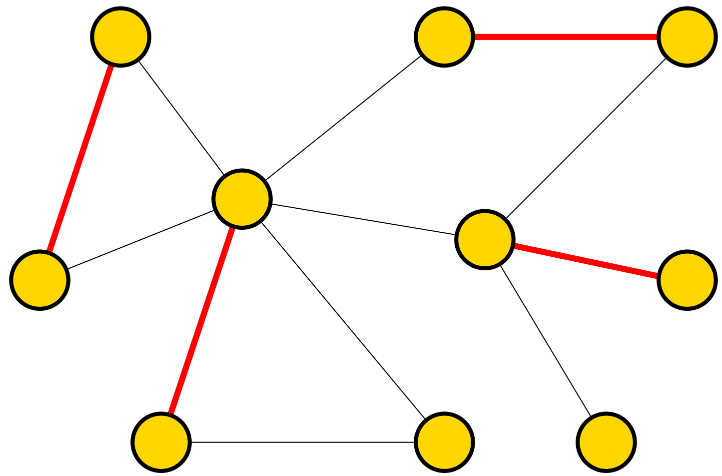


What is known:

- $1 - \epsilon$ approximation in $\tilde{O}(n)$ space and $f(\epsilon)$ passes [McGregor 2005] [Eggert, Kliemann, Munstermann, Srivastav 2012] [Ahn, Guha 2011]

Matchings

The Maximum Matching Problem:

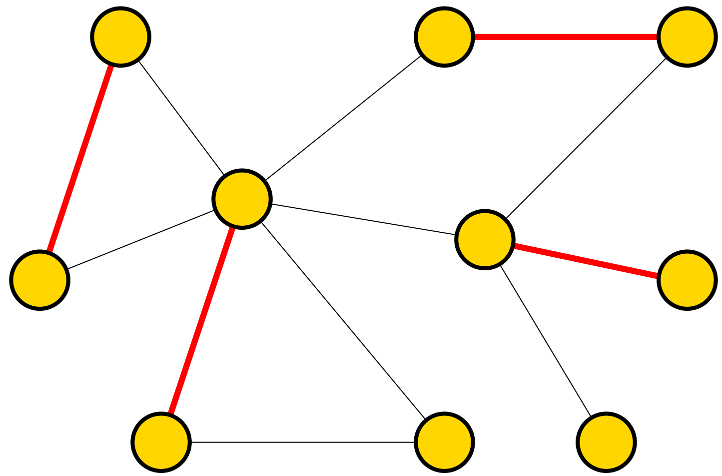


What is known:

- $1-\epsilon$ approximation in $\tilde{O}(n)$ space and $f(\epsilon)$ passes [McGregor 2005] [Eggert, Kliemann, Munstermann, Srivastav 2012] [Ahn, Guha 2011]
- $n^{1+\Omega(1/\log\log n)}$ lower bound for $(1-e^{-1}+\delta)$-approximation in one pass [Goel, Kapralov, Khanna 2012] [Kapralov 2012]

Matchings

The Maximum Matching Problem:

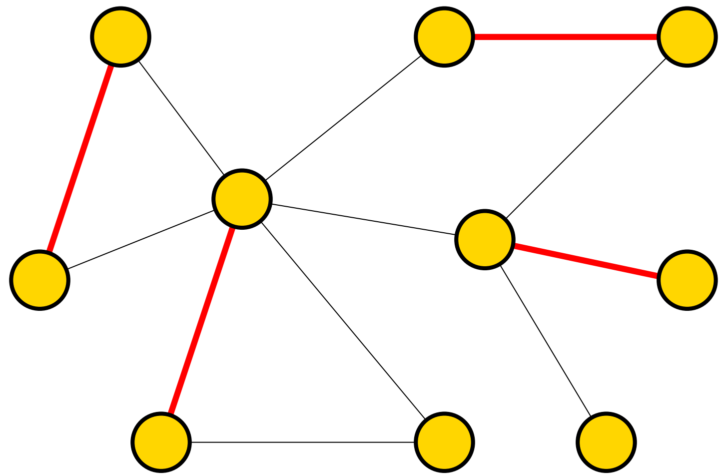


What is known:

- Great open question: Can obtain $1/2 + \Omega(1)$ approximation in one pass with $\tilde{O}(n)$ space?

Matchings

The Maximum Matching Problem:

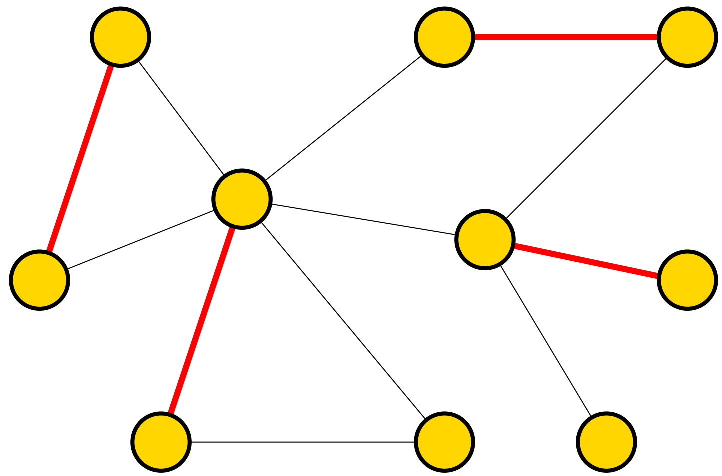


What is known:

- Great open question: Can obtain $1/2 + \Omega(1)$ approximation in one pass with $\tilde{O}(n)$ space?
  - Two passes are enough [Konrad, Magniez, Mathieu 2011]
  - Possible for random ordering [Konrad, Magniez, Mathieu 2011]

Matchings

The Maximum Matching Problem:

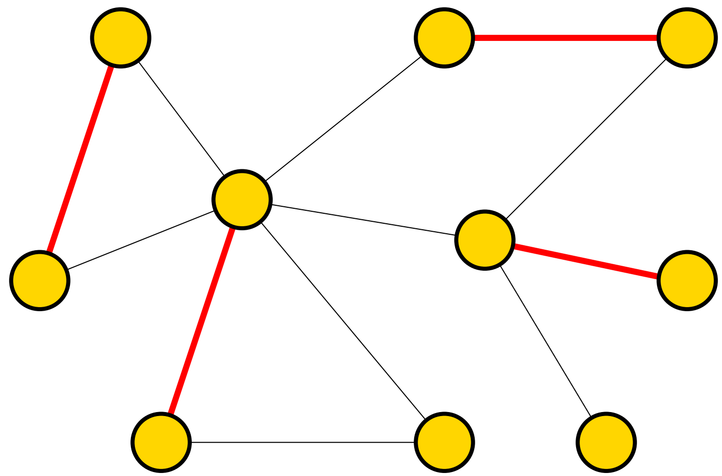


What is known:

- Feigenbaum, Kannan, McGregor, Suri, Zhang (2004): $\Omega(n^2)$ lower bound for exact matching in one pass

Matchings

The Maximum Matching Problem:

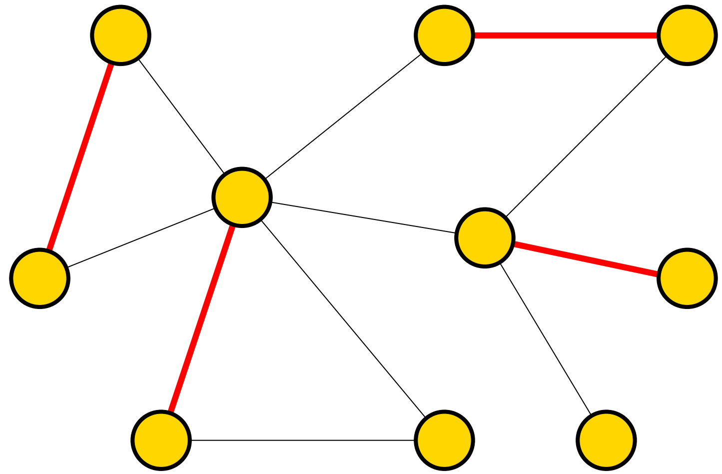


What is known:

- Feigenbaum, Kannan, McGregor, Suri, Zhang (2004): $\Omega(n^2)$ lower bound for exact matching in one pass
- $\Omega(n)$ lower bounds relatively easy

Matchings

The Maximum Matching Problem:

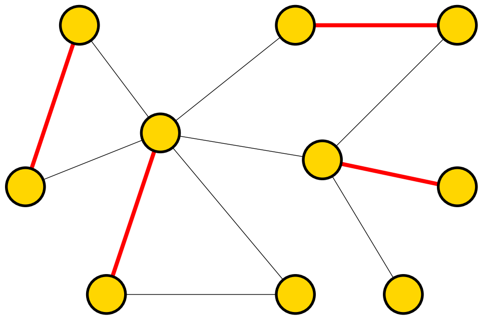


What is known:

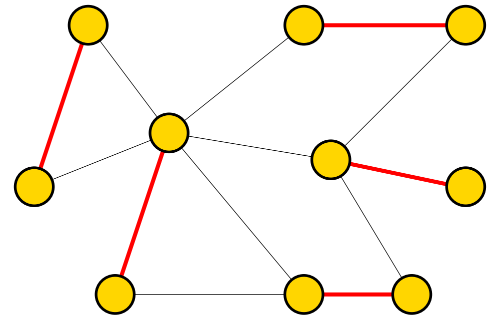
- Feigenbaum, Kannan, McGregor, Suri, Zhang (2004): $\Omega(n^2)$ lower bound for exact matching in one pass
- $\Omega(n)$ lower bounds relatively easy
- No $\omega(n \log n)$ lower bound known even for computing an exact matching in multiple passes

Maximum Matching

Problem: Is there a perfect matching?

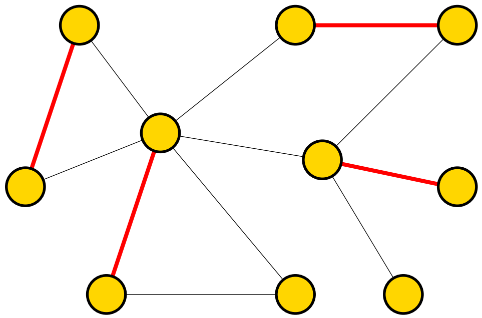


vs.

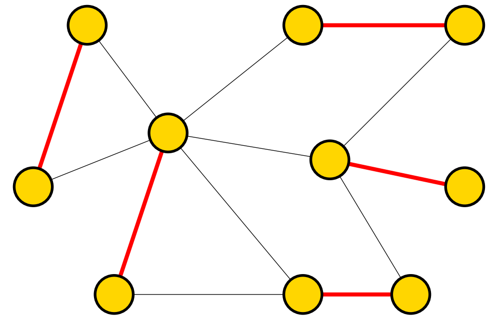


Maximum Matching

Problem: Is there a perfect matching?



vs.

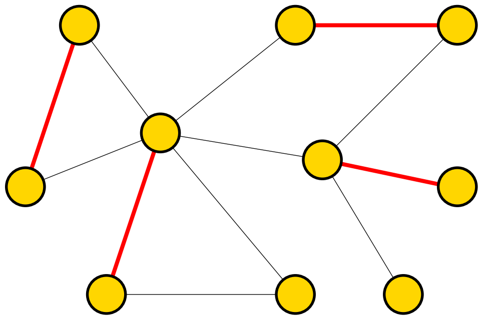


Our Result:

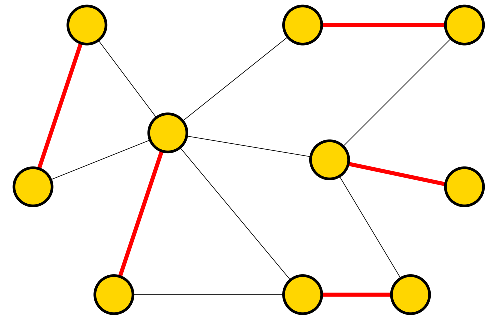
$\sim n^{1+\Omega(1/p)}$ space required for p passes

Maximum Matching

Problem: Is there a perfect matching?



vs.



Our Result:

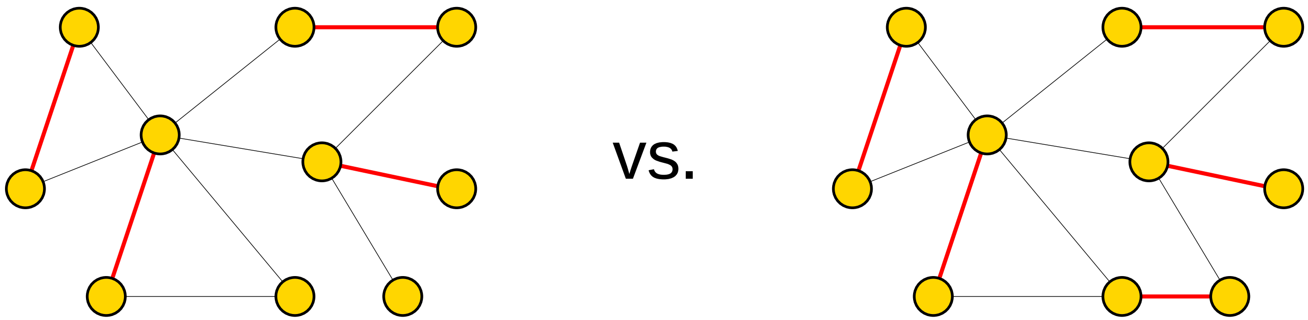
$\sim n^{1+\Omega(1/p)}$ space required for p passes

Implies lower bounds for:

- What is the size of maximum matching?

Maximum Matching

Problem: Is there a perfect matching?



Our Result:

$\sim n^{1+\Omega(1/p)}$ space required for p passes

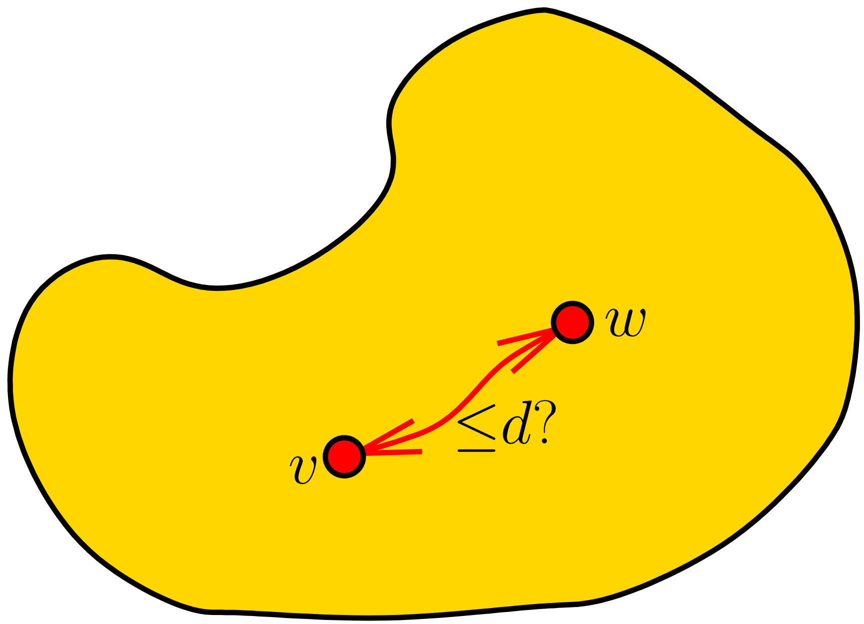
Implies lower bounds for:

- What is the size of maximum matching?
- Find maximum matching

Shortest Path(s)

Problem:

Are two vertices v and w at distance $2(p+1)$?



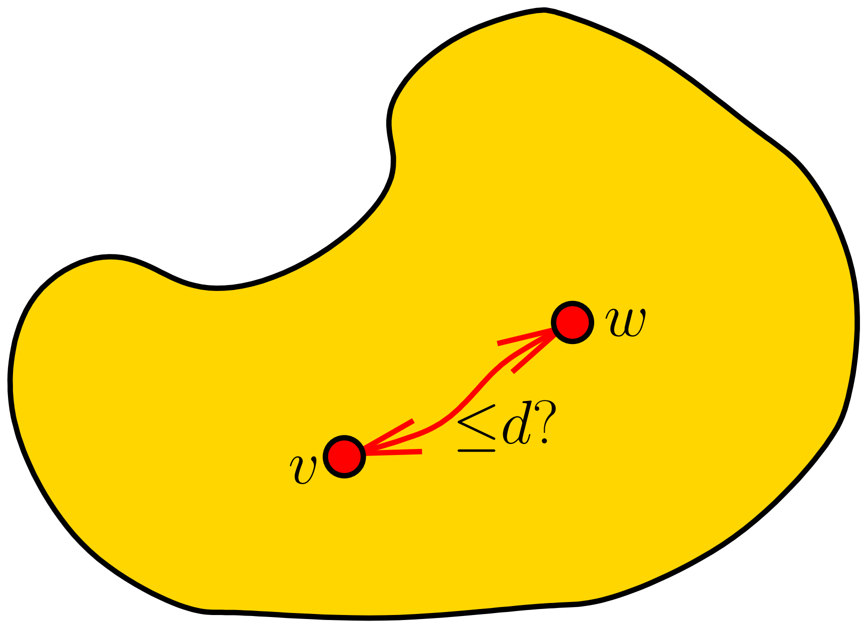
Shortest Path(s)

Problem:

Are two vertices v and w at distance $2(p+1)$?

Our Result:

$\sim n^{1+\Omega(1/p)}$ space required for p passes



Shortest Path(s)

Problem:

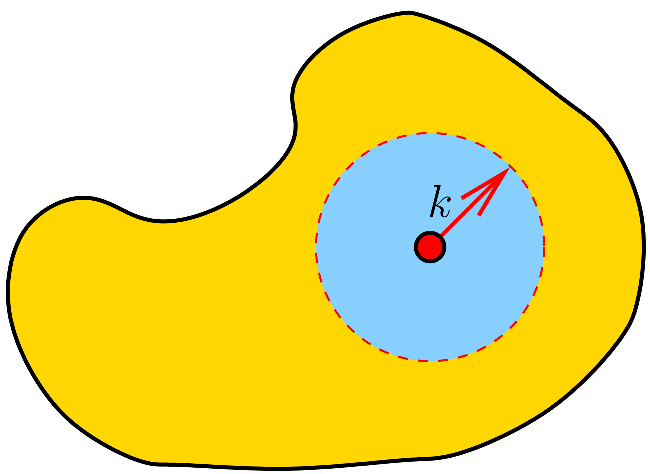
Are two vertices v and w at distance $2(p+1)$?

Our Result:

$\sim n^{1+\Omega(1/p)}$ space required for p passes

Previous Results [Feigenbaum, Kannan, McGregor, Suri, Zhang 2005]:

- Computing the first $k = O(1)$ layers of BFS tree in $<k/2$ passes requires $\Omega(n^{1+1/k}/(\log n)^{1/k})$ space



Shortest Path(s)

Problem:

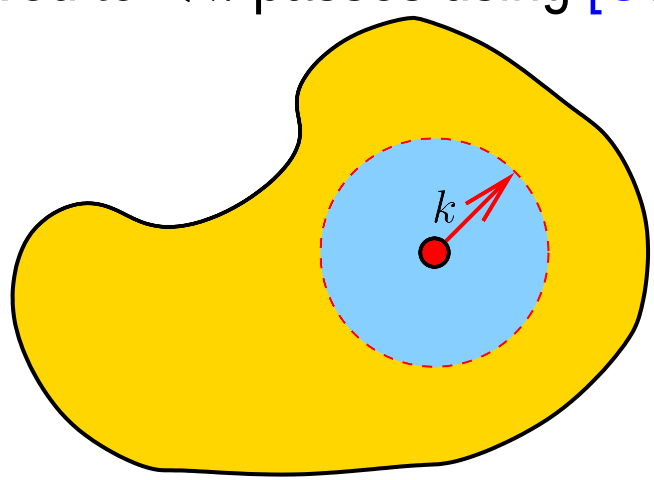
Are two vertices v and w at distance $2(p+1)$?

Our Result:

$\sim n^{1+\Omega(1/p)}$ space required for p passes

Previous Results [Feigenbaum, Kannan, McGregor, Suri, Zhang 2005]:

- Computing the first $k = O(1)$ layers of BFS tree in $< k/2$ passes requires $\Omega(n^{1+1/k}/(\log n)^{1/k})$ space
  - Can be improved to $< k$ passes using [Guha, McGregor 2007]



Shortest Path(s)

Problem:

Are two vertices v and w at distance $2(p+1)$?

Our Result:

$\sim n^{1+\Omega(1/p)}$ space required for p passes

Previous Results [Feigenbaum, Kannan, McGregor, Suri, Zhang 2005]:

- Computing the first $k = O(1)$ layers of BFS tree in $<k/2$ passes requires $\Omega(n^{1+1/k}/(\log n)^{1/k})$ space
  - Can be improved to $< k$ passes using [Guha, McGregor 2007]
- $(2t+1)$-spanner construction in $O(tn^{1+1/t}\log^2 n)$ space and one pass

Shortest Path(s)

Problem:

Are two vertices v and w at distance $2(p+1)$?

Our Result:

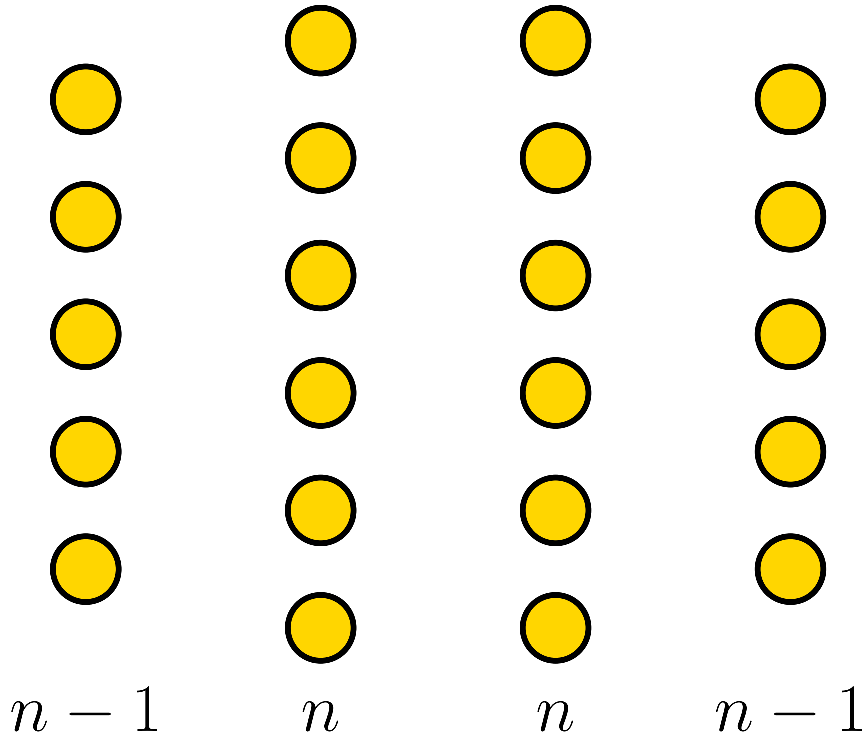
$\sim n^{1+\Omega(1/p)}$ space required for p passes

Previous Results [Feigenbaum, Kannan, McGregor, Suri, Zhang 2005]:

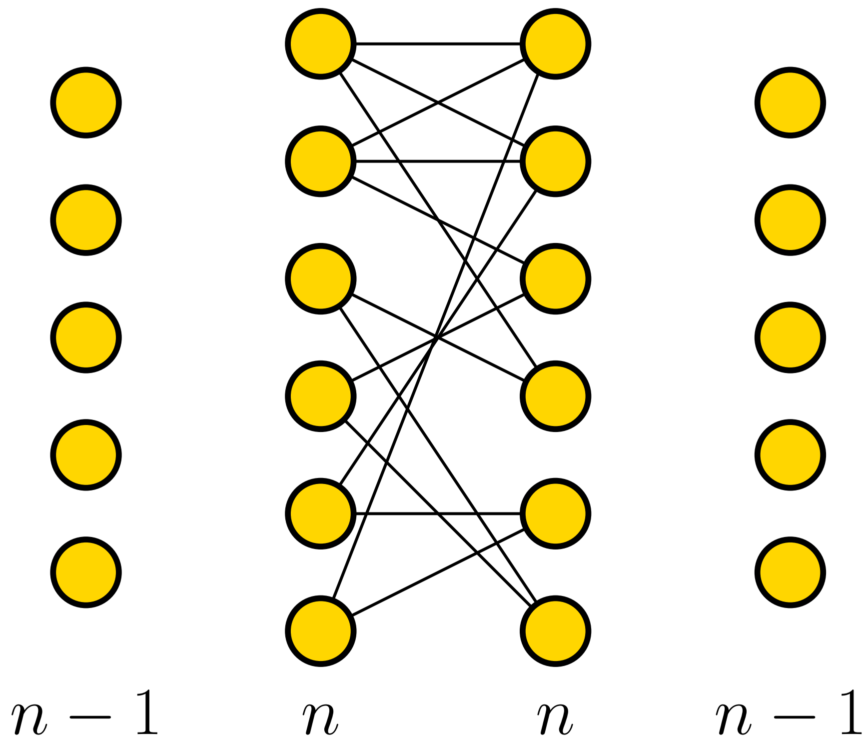
- Computing the first $k = O(1)$ layers of BFS tree in $<k/2$ passes requires $\Omega(n^{1+1/k}/(\log n)^{1/k})$ space
  - Can be improved to $< k$ passes using [Guha, McGregor 2007]
- $(2t+1)$-spanner construction in $O(tn^{1+1/t}\log^2 n)$ space and one pass
- t-approximation of distance between two nodes in one pass requires $\Omega(n^{1+1/t})$ space

Warmup:
One-Pass Lower Bound
[Feigenbaum et al. 2004]

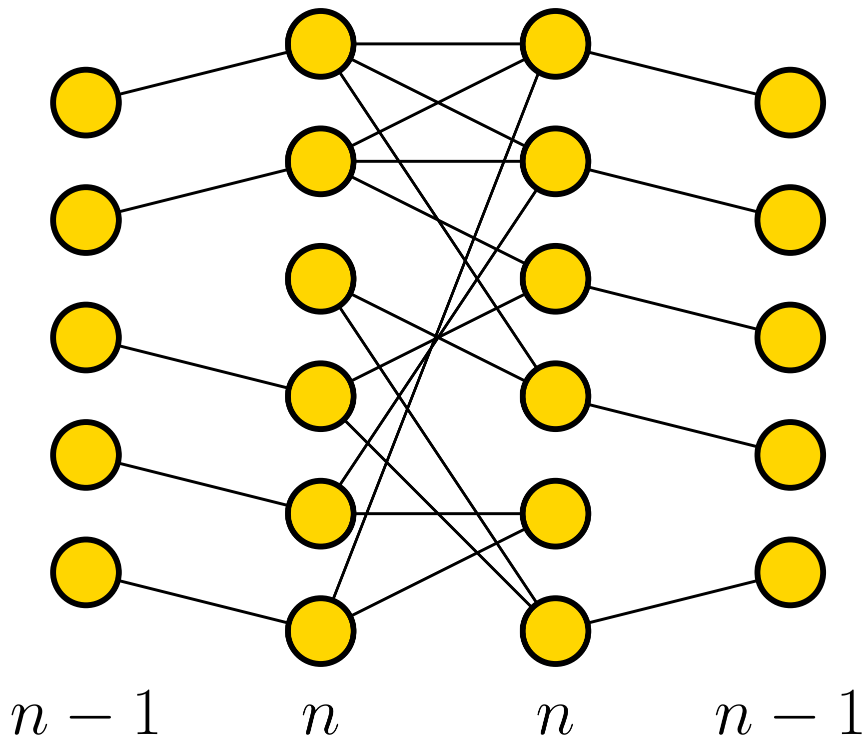
Construction for Perfect Matching



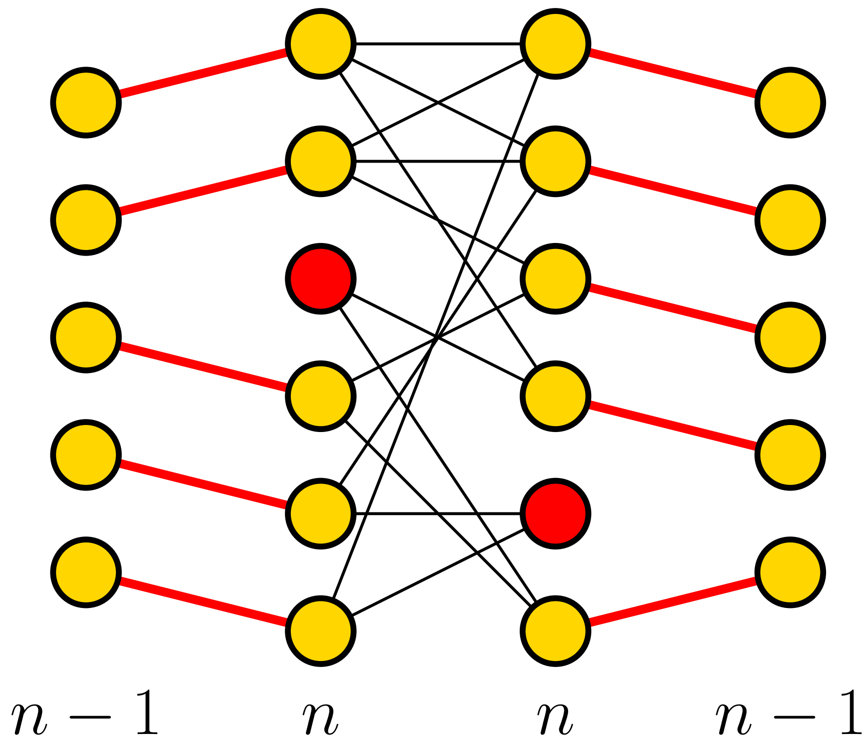
Construction for Perfect Matching



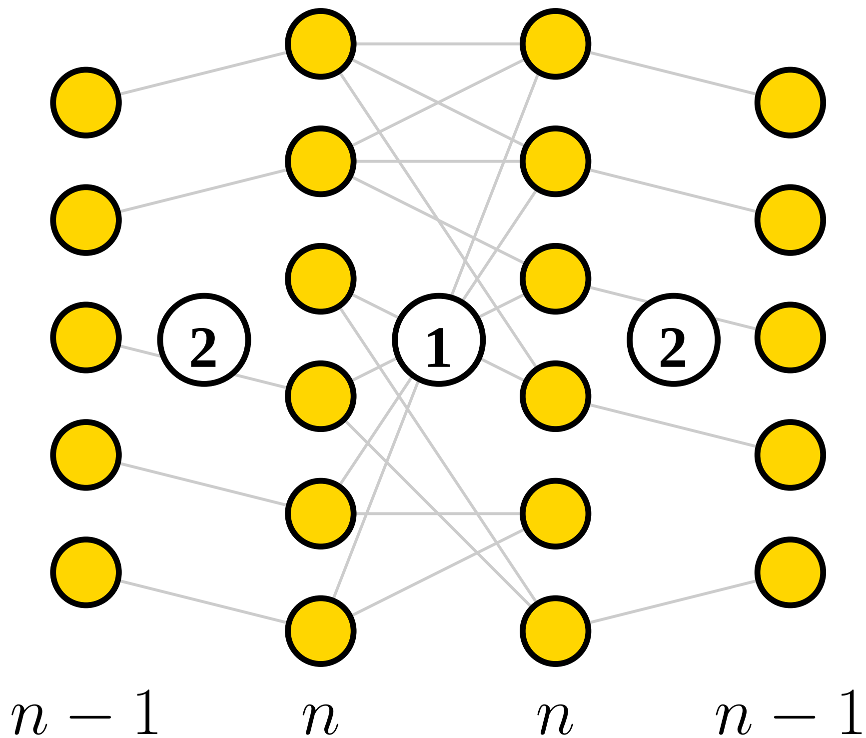
Construction for Perfect Matching



Construction for Perfect Matching

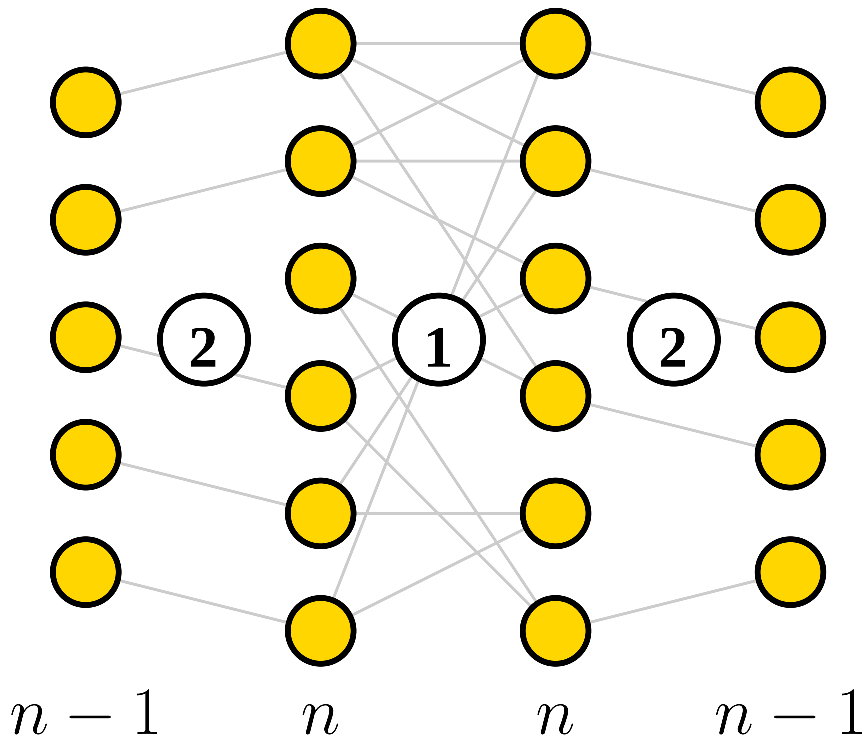


Construction for Perfect Matching



Stream = ① ②

Construction for Perfect Matching



Stream = ① ②

Lower bound of $\Omega(n^2)$ via indexing

Alice	$\Rightarrow$	Bob
$A[1 \ldots n^2]$		x

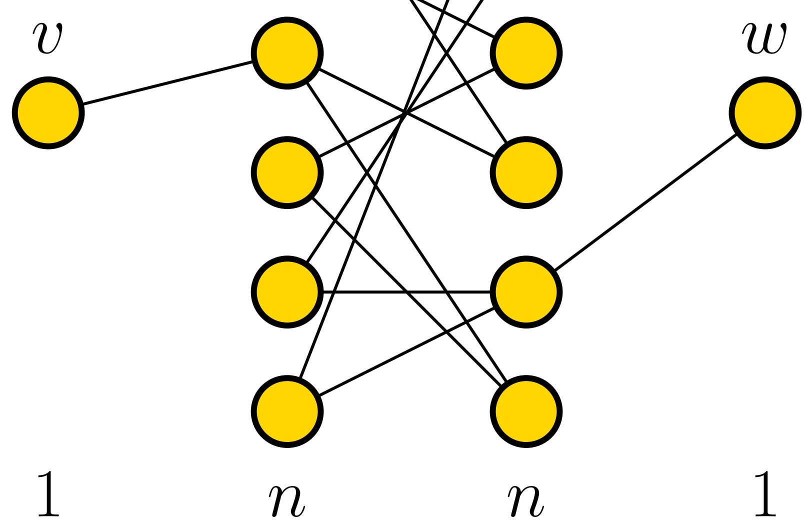
Bob's task: output $A[x]$

Construction for Shortest Path

Approximation better then $5/3$ requires $\Omega(n^2)$ space

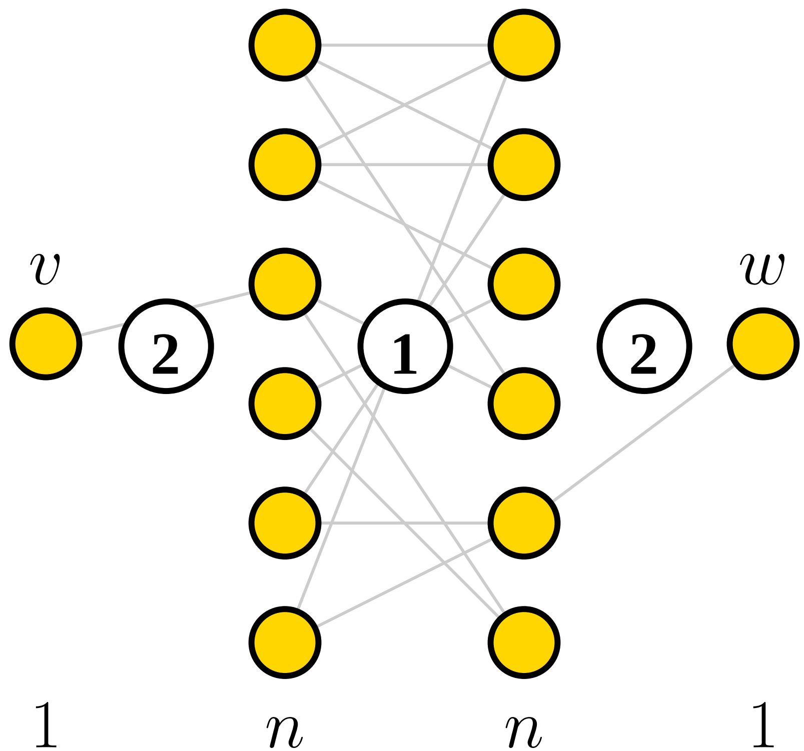
Construction for Shortest Path

Approximation better then $5/3$ requires $\Omega(n^2)$ space



Construction for Shortest Path

Approximation better then $5/3$ requires $\Omega(n^2)$ space

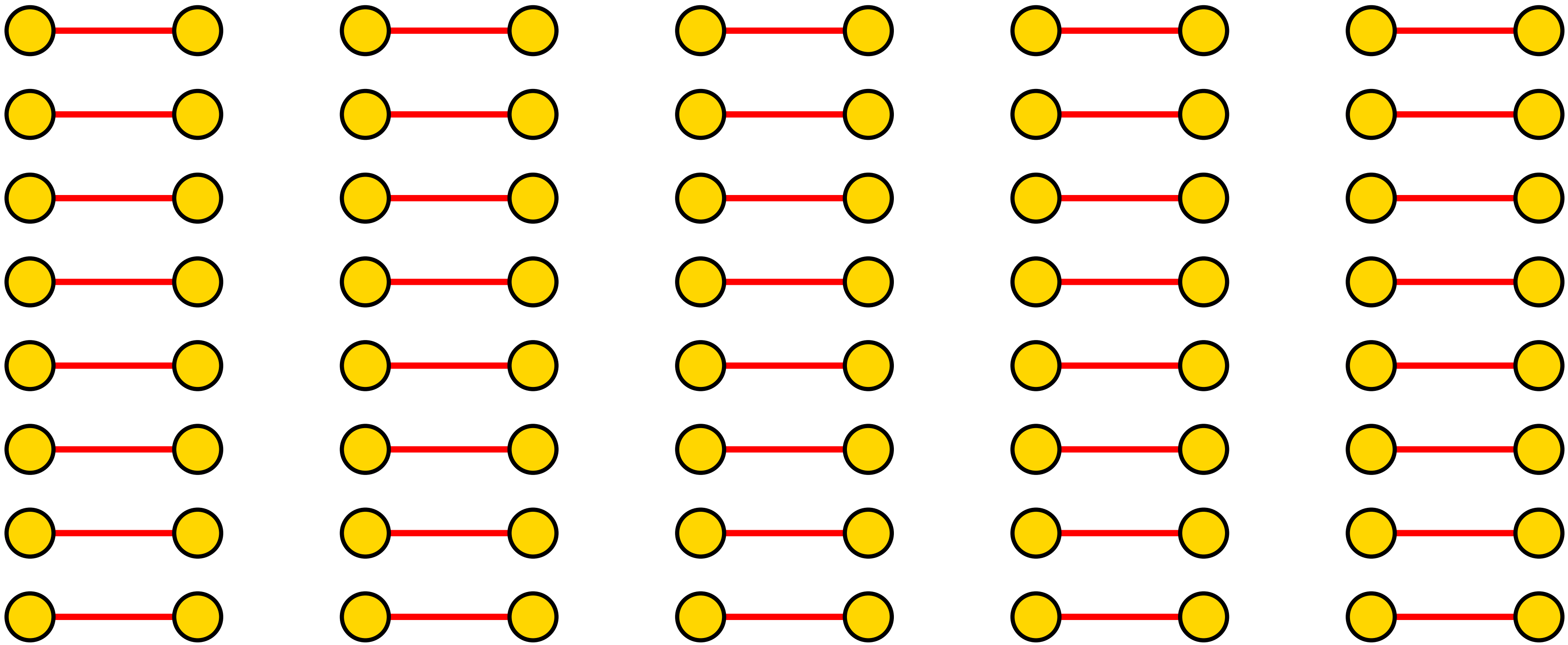


Stream $=$ ① ②

Hard Instance for Multiple Passes

Construction for Perfect Matching

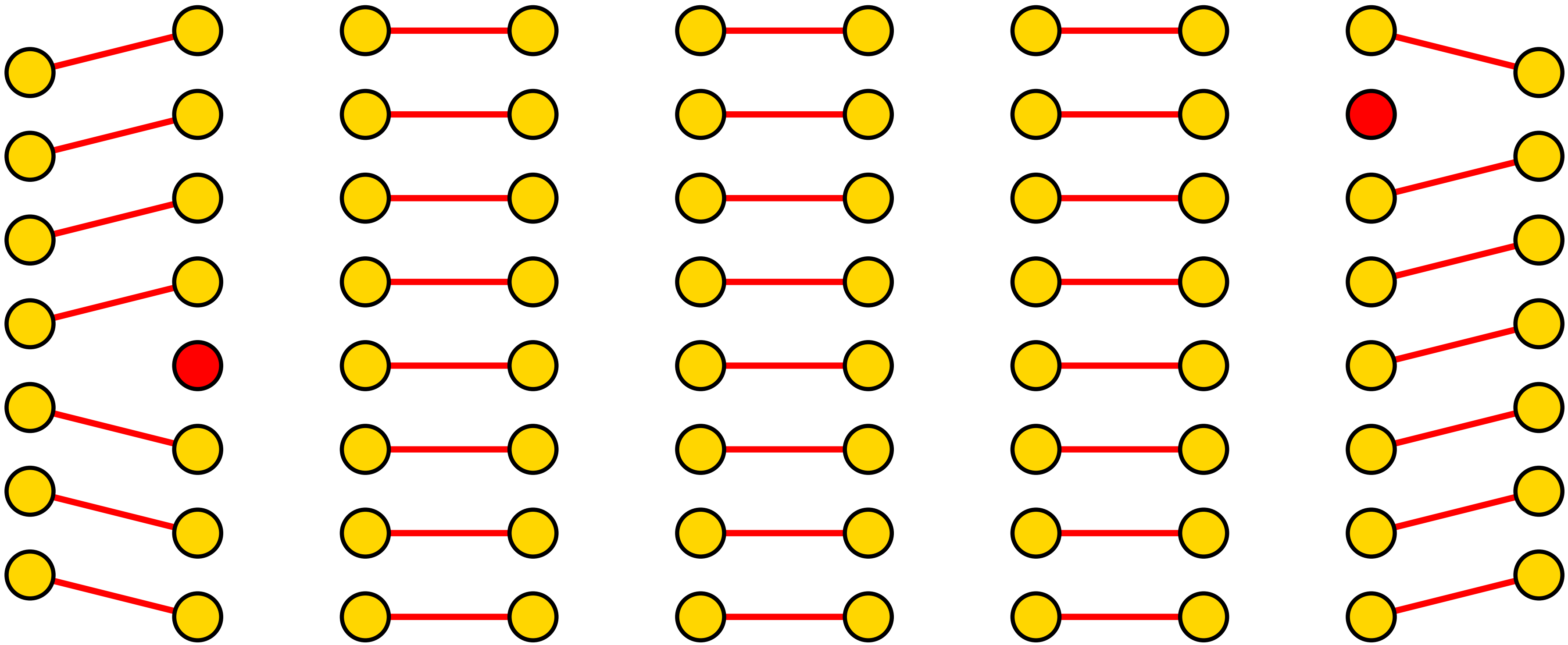
Is there a perfect matching?



$\Theta(1)$ columns Each column $\Theta(n)$ rows

Construction for Perfect Matching

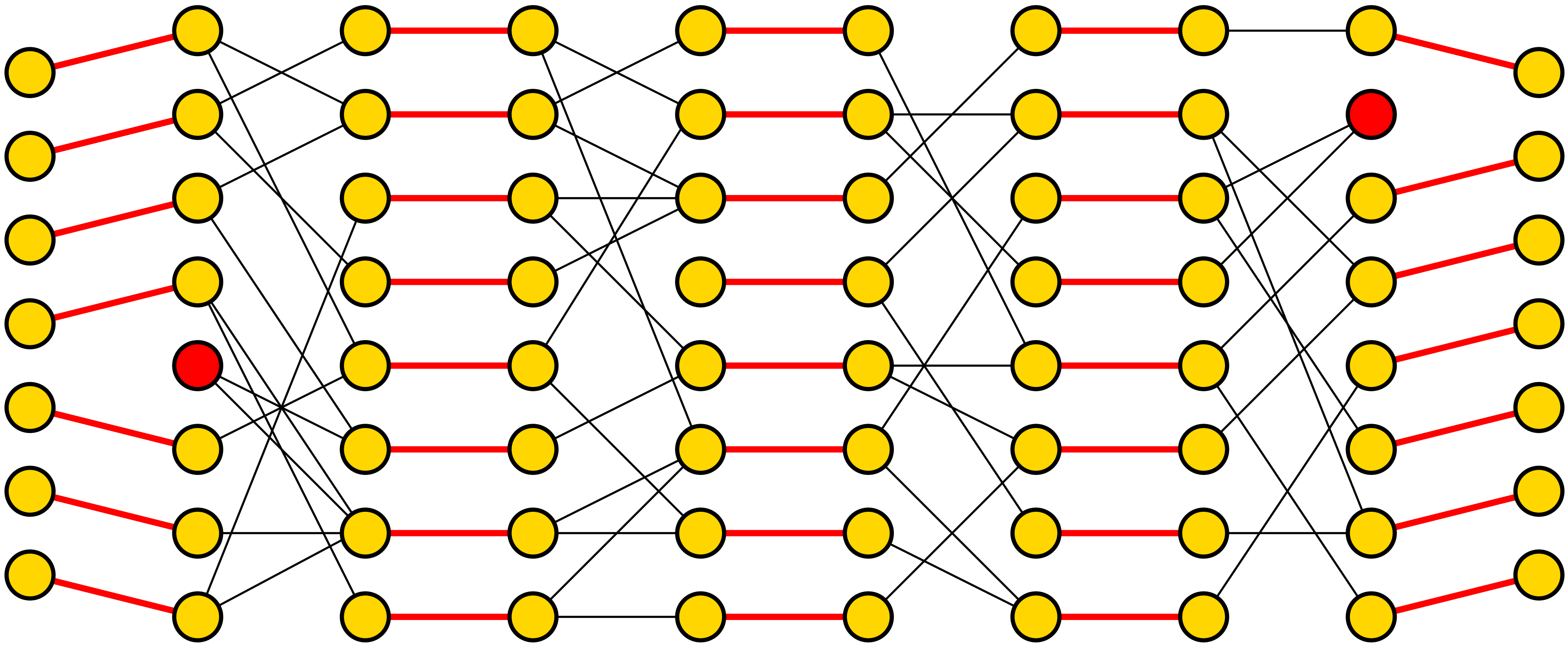
Is there a perfect matching?



$\Theta(1)$ columns Each column $\Theta(n)$ rows

Construction for Perfect Matching

Is there a perfect matching?

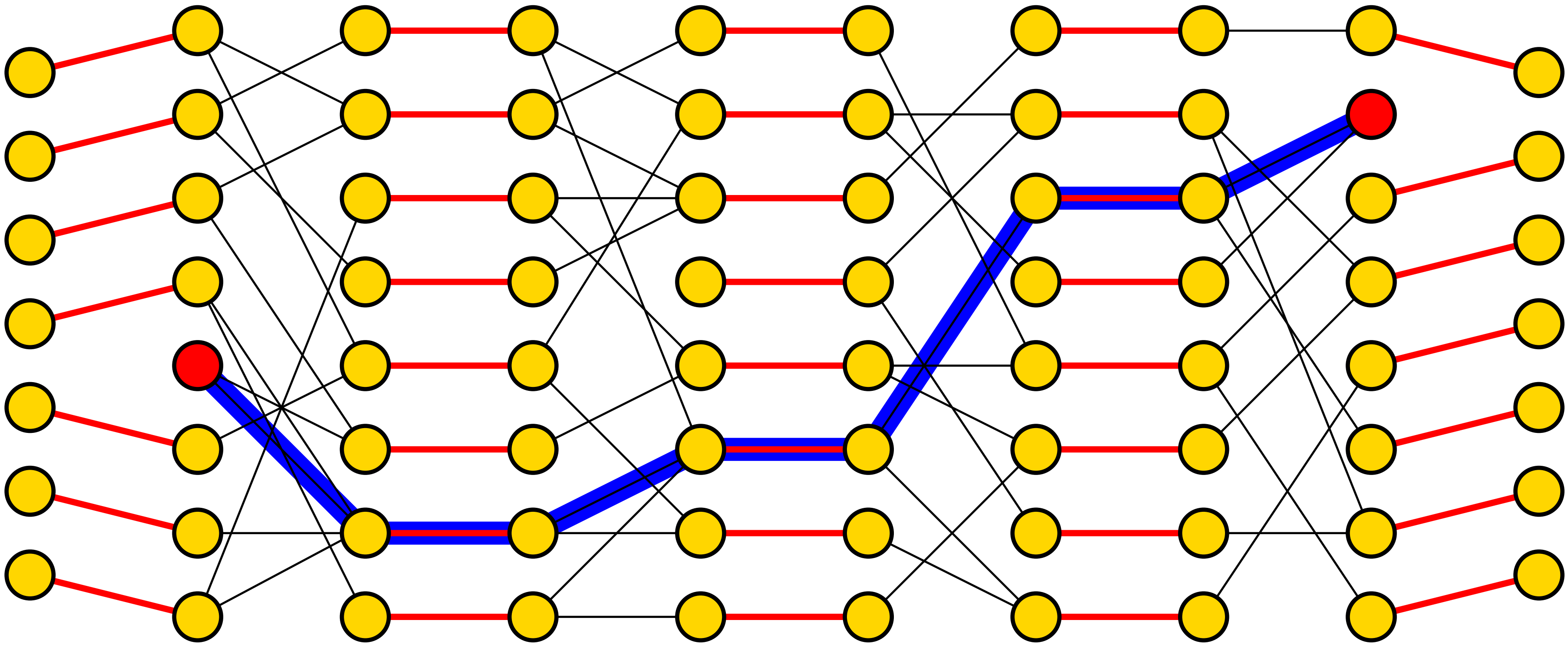


$\Theta(1)$ columns

Each column $\Theta(n)$ rows

Construction for Perfect Matching

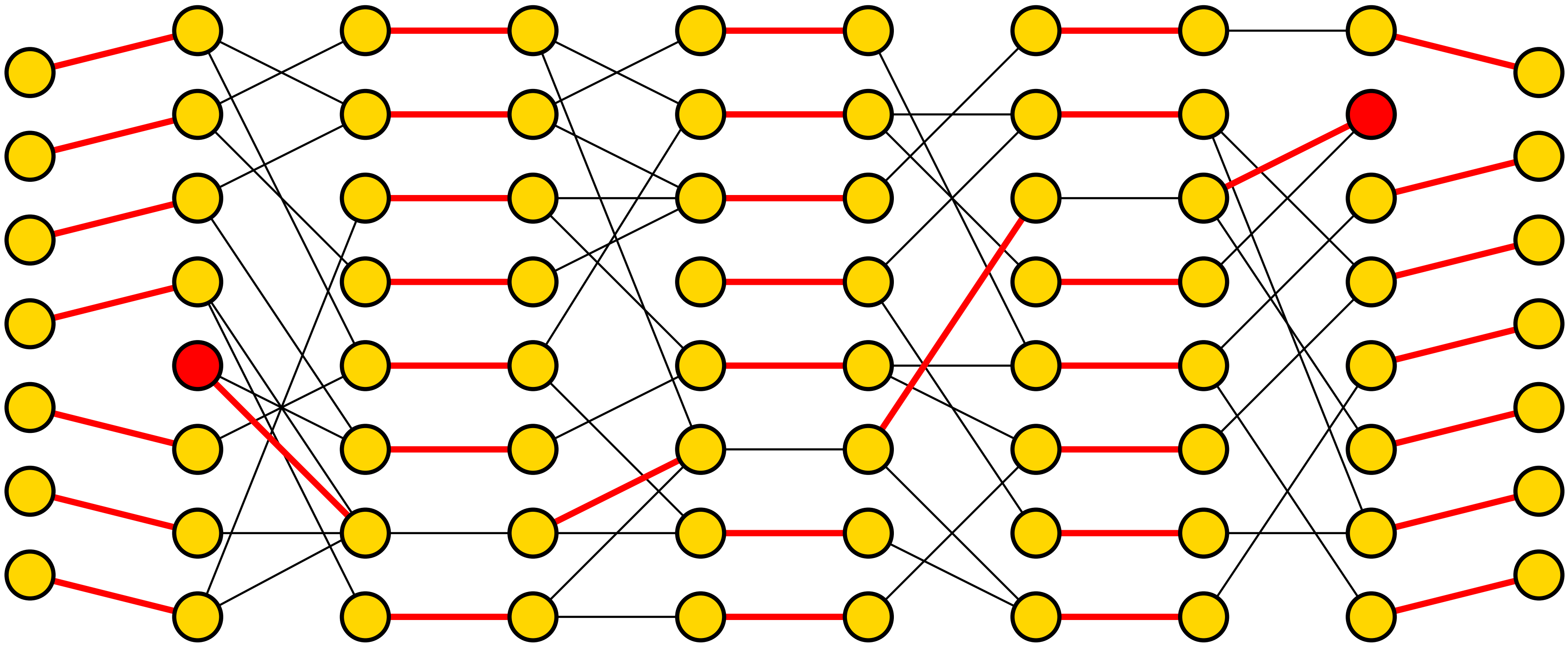
Is there a perfect matching?



$\Theta(1)$ columns　　　　Each column $\Theta(n)$ rows

Construction for Perfect Matching

Is there a perfect matching?

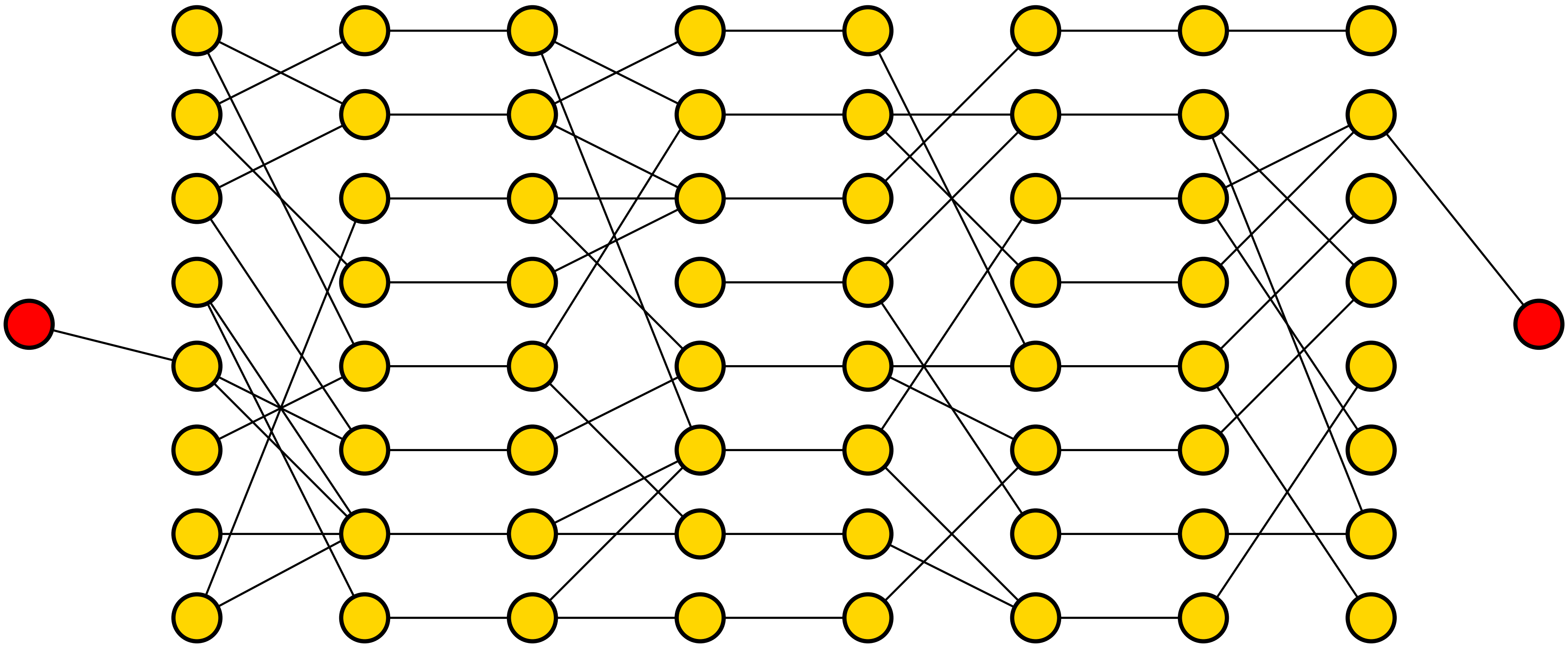


$\Theta(1)$ columns

Each column $\Theta(n)$ rows

Construction for Perfect Matching

Is there a path of length 9 between red nodes?

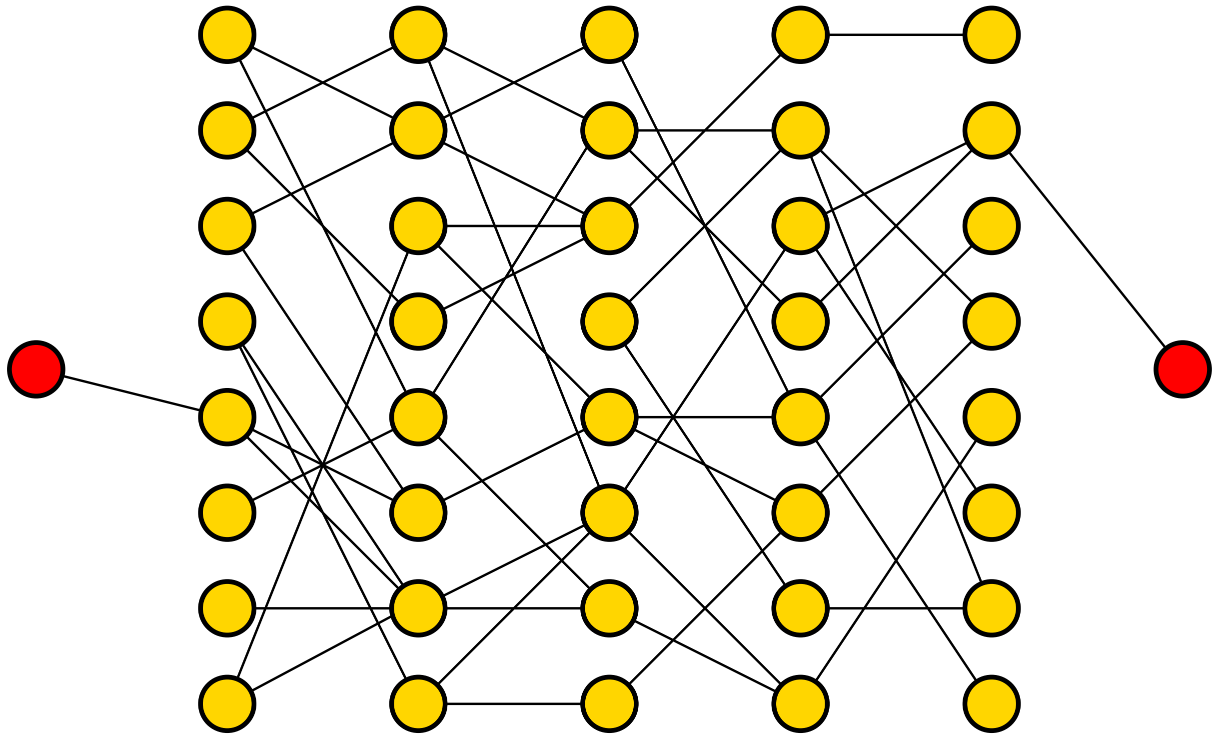


$\Theta(1)$ columns

Each column $\Theta(n)$ rows

Construction for Perfect Matching

Is there a path of length 6 between red nodes?

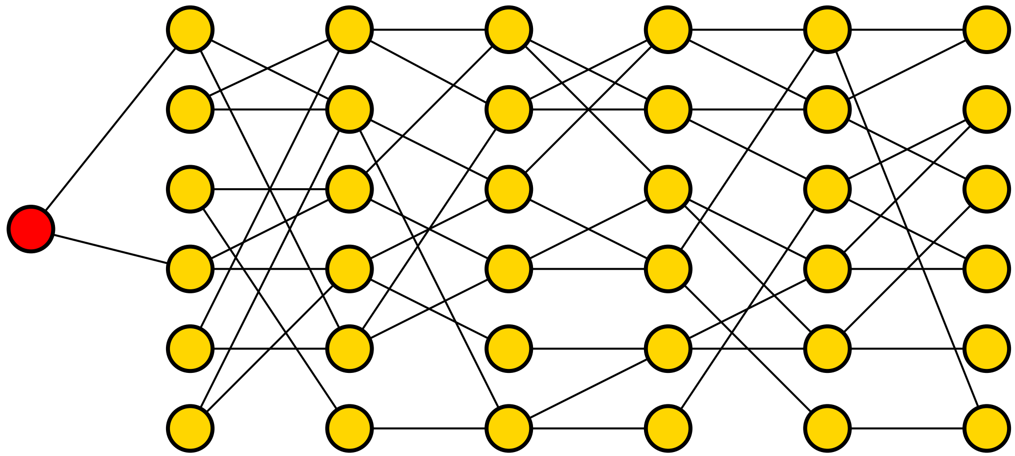


$\Theta(1)$ columns

Each column $\Theta(n)$ rows

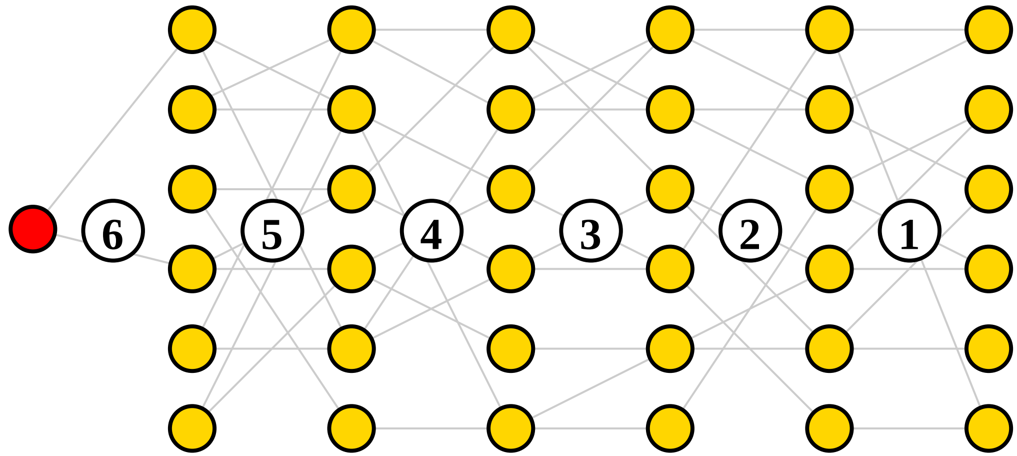
Quick Comparison to FKMSZ'05

With multiplayer pointer chasing from [Guha, McGregor 2007]:



Quick Comparison to FKMSZ'05

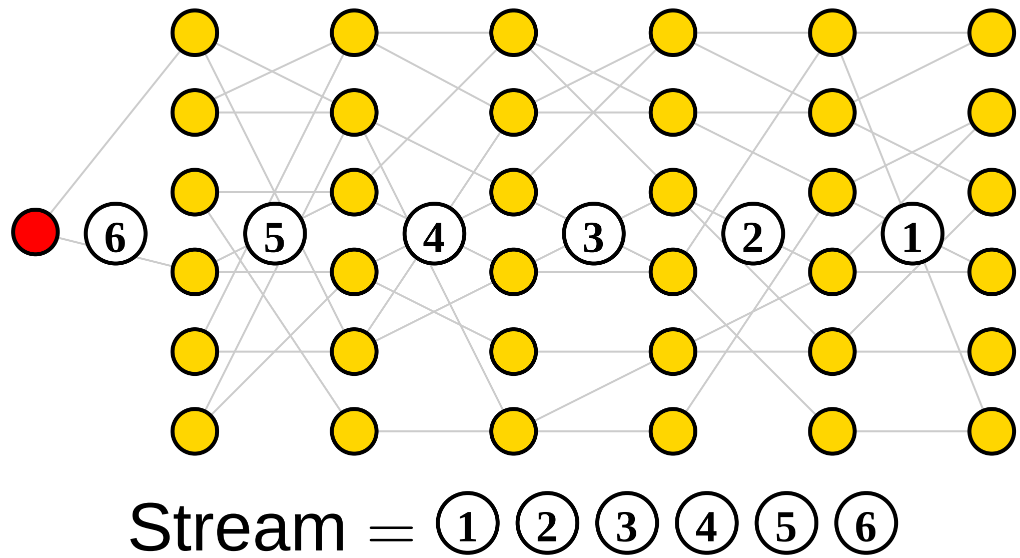
With multiplayer pointer chasing from [Guha, McGregor 2007]:



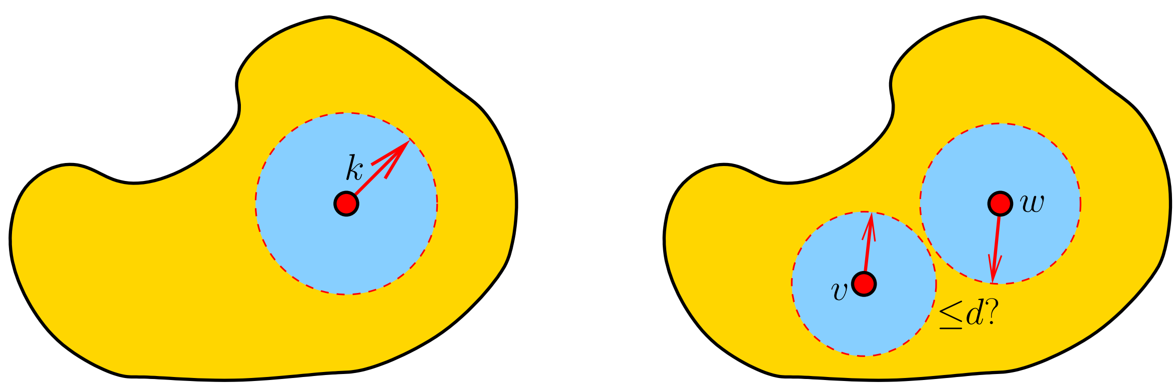
Stream = ① ② ③ ④ ⑤ ⑥

Quick Comparison to FKMSZ’05

With multiplayer pointer chasing from [Guha, McGregor 2007]:



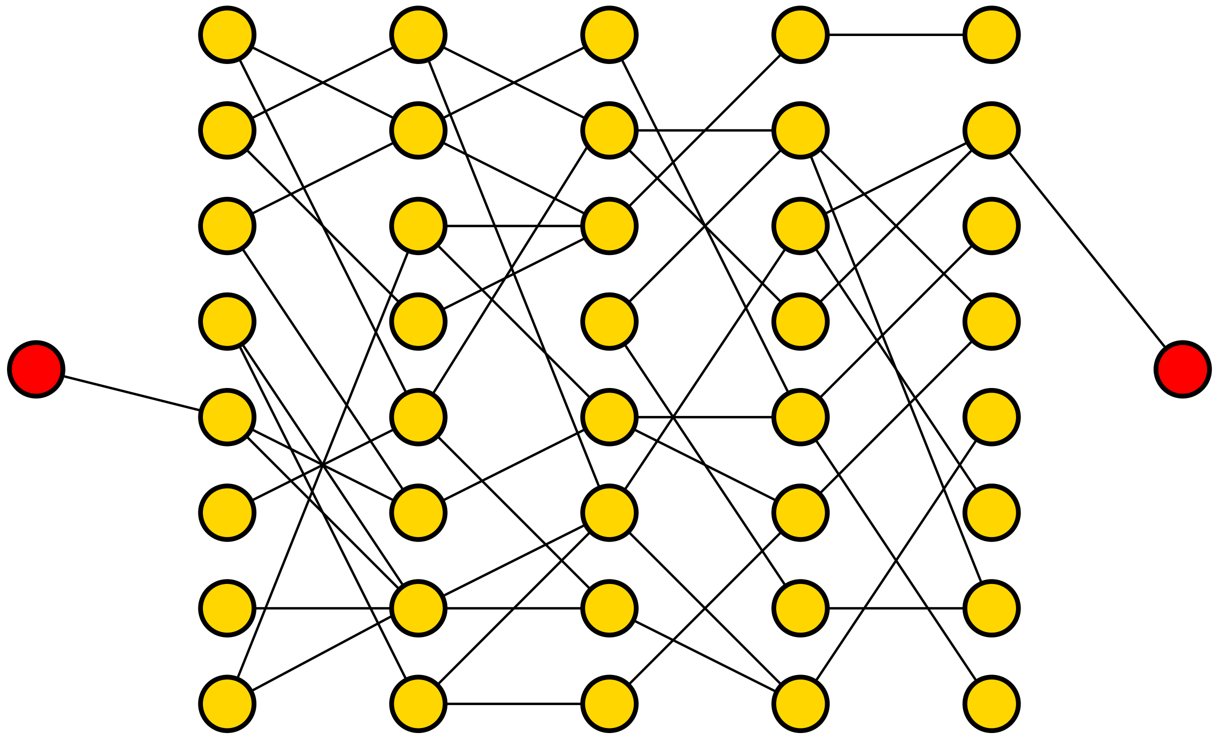
Our problem: Fewer passes suffice



[FKMSZ’05] Here

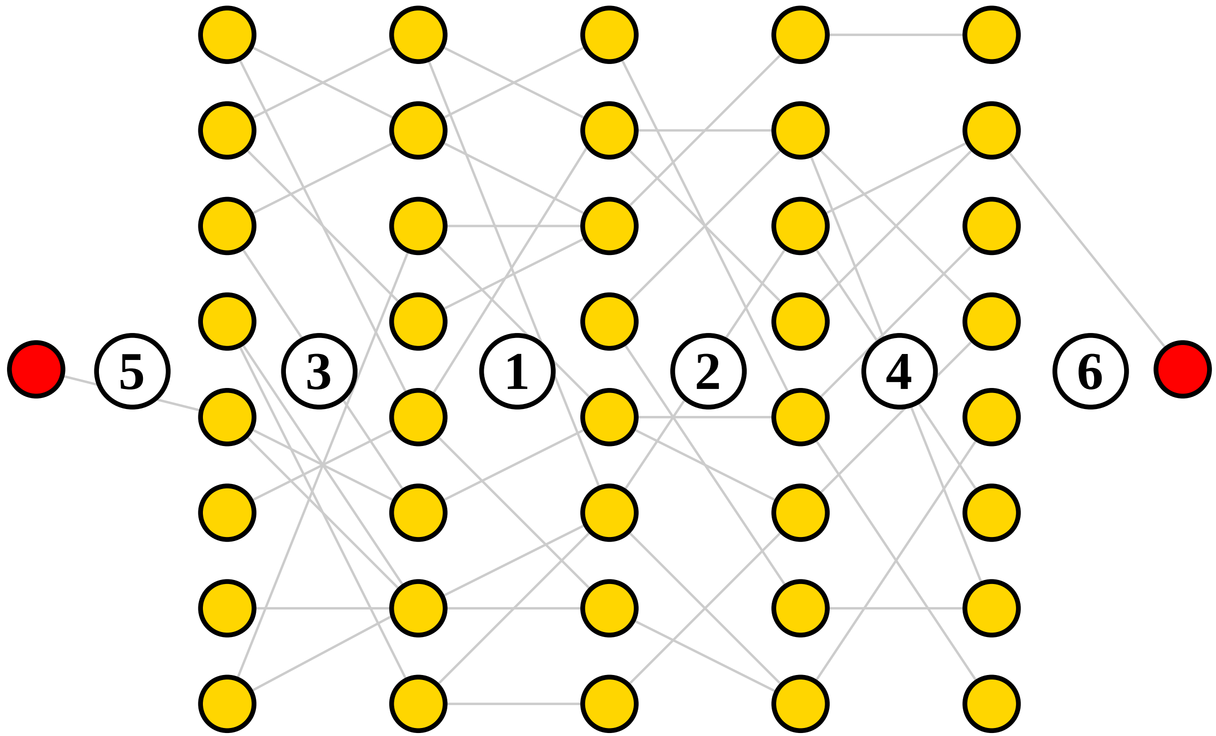
Our Stream Ordering

Is there a path of length 6 between red nodes?



Our Stream Ordering

Is there a path of length 6 between red nodes?



Stream = ① ② ③ ④ ⑤ ⑥

Proof Sketch

Proof Sketch

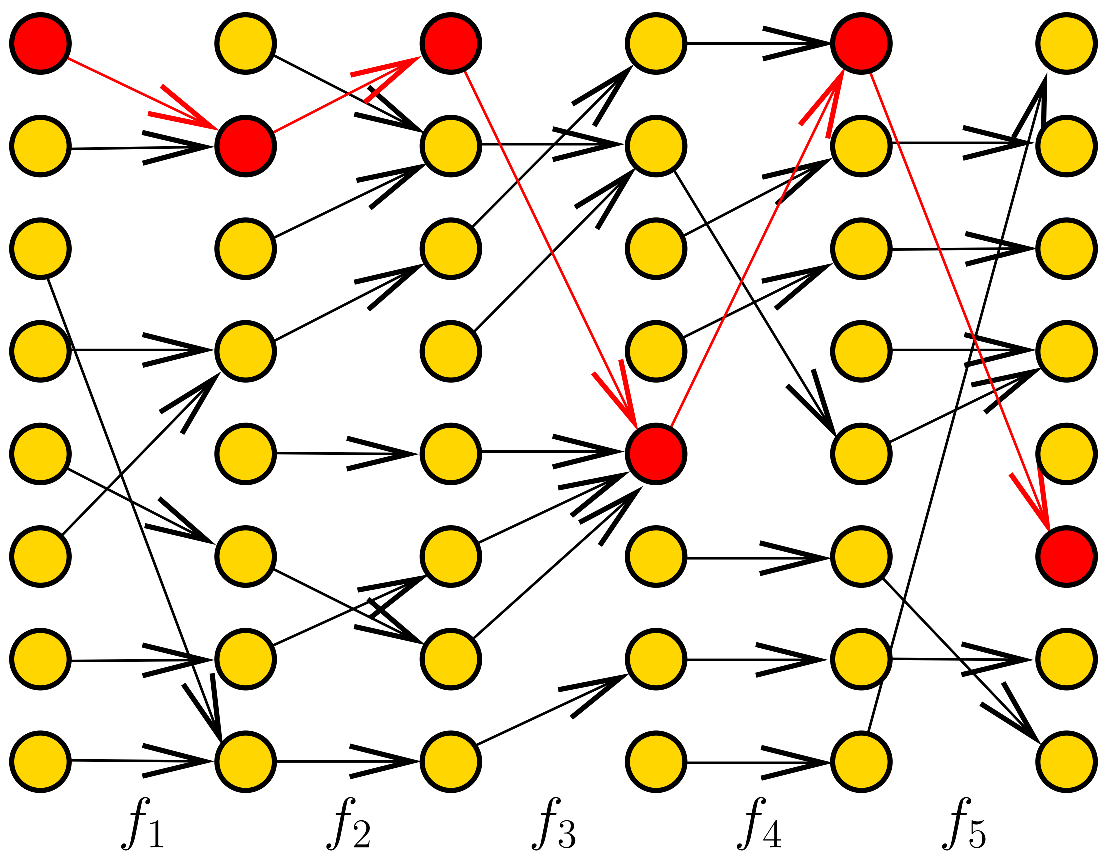
Quick note on our protocols:

- Randomized
- Private randomness
- Public communication

Important Problem: Pointer Chasing

Definition:

- Input: k functions $f_i : [n] \to [n]$
- Goal: Compute $f_k(f_{k-1}(\dots f_2(f_1(1))\dots))$



Important Problem: Pointer Chasing

Definition:

- Input: k functions $f_i : [n] \to [n]$
- Goal: Compute $f_k(f_{k-1}(\ldots f_2(f_1(1))\ldots))$

Two-player version:

- What players have:

Alice	Bob
$f_2, f_4, f_6, \ldots$	$f_1, f_3, f_5, \ldots$

- Alice speaks first

Important Problem: Pointer Chasing

Definition:

- Input: k functions $f_i : [n] \to [n]$
- Goal: Compute $f_k(f_{k-1}(\ldots f_2(f_1(1))\ldots))$

Two-player version:

- What players have:

Alice	Bob
$f_2, f_4, f_6, \ldots$	$f_1, f_3, f_5, \ldots$

- Alice speaks first
- Nisan, Wigderson (1993):

Computing in less then $k = \Theta(1)$ messages
of communication requires $\Omega(n)$ communication

Important Problem: Pointer Chasing

Definition:

- Input: k functions $f_i : [n] \to [n]$
- Goal: Compute $f_k(f_{k-1}(\ldots f_2(f_1(1))\ldots))$

k-player version:

- What players have:

Player 1	Player 2	...	Player $k-1$	Player k
f_k	f_{k-1}	...	f_2	f_1

- Each round: players speak in order Player 1 through Player k

Important Problem: Pointer Chasing

Definition:

- Input: k functions $f_i : [n] \to [n]$
- Goal: Compute $f_k(f_{k-1}(\dots f_2(f_1(1))\dots))$

k-player version:

- What players have:

Player 1	Player 2	...	Player $k-1$	Player k
f_k	f_{k-1}	...	f_2	f_1

- Each round: players speak in order Player 1 through Player k
- Guha, McGregor (2007):

Computing in less then $k = \Theta(1)$ rounds
requires $\Omega(n)$ communication

Feigenbaum et al. (2005)

Their Problem:

Compute p levels of BFS tree from v

Important question:

Why can't reduce to FKMSZ'05?

Feigenbaum et al. (2005)

Their Problem:

Compute p levels of BFS tree from v

Important question:

Why can't reduce to FKMSZ'05?

Sketch of their proof:

- Take Nisan-Wigderson (1993) communication lower bound for pointer chasing

Feigenbaum et al. (2005)

Their Problem:

Compute p levels of BFS tree from v

Important question:

Why can't reduce to FKMSZ'05?

Sketch of their proof:

- Take Nisan-Wigderson (1993) communication lower bound for pointer chasing
- Apply direct sum theorem of Jain, Radhakrishnan, Sen (2003): Solving k instances requires k times more communication

Feigenbaum et al. (2005)

Their Problem:

Compute p levels of BFS tree from v

Important question:

Why can't reduce to FKMSZ'05?

Sketch of their proof:

- Take Nisan-Wigderson (1993) communication lower bound for pointer chasing
- Apply direct sum theorem of Jain, Radhakrishnan, Sen (2003): Solving k instances requires k times more communication
- If can compute BFS of graph of degree $k = n^{\Theta(1/p)}$, then can solve k instances of pointer chasing

Complexity Measures

Functions of the input size:

- Information Cost: $\mathrm{ICost}_\mu(\Pi) = I(X : \Pi(X))$ where
  - $X =$ input selected from μ
  - $\Pi(X)$ transcript of Π on X

Complexity Measures

Functions of the input size:

- Information Cost: $\mathrm{ICost}_{\mu}(\Pi) = I(X : \Pi(X))$
  where
  - $X =$ input selected from μ
  - $\Pi(X)$ transcript of Π on X
- Information Complexity: $\mathrm{IC}_{\mu,\delta}(P) = \inf_{\Pi} \mathrm{ICost}_{\mu}(\Pi)$
  where
  - infimum is taken over protocols Π that solve problem P with probability $1 - \delta$

Complexity Measures

Functions of the input size:

- Information Cost: $\mathrm{ICost}_\mu(\Pi) = I(X : \Pi(X))$
  where
  - $X =$ input selected from μ
  - $\Pi(X)$ transcript of Π on X
- Information Complexity: $\mathrm{IC}_{\mu,\delta}(P) = \inf_\Pi \mathrm{ICost}_\mu(\Pi)$
  where
  - infimum is taken over protocols Π that solve problem P with probability $1 - \delta$
- Communication Complexity: $\mathrm{CC}_\delta(P) = \inf_\Pi \max_X |\Pi(X)|$
  where
  - infimum is taken over protocols Π that solve problem P on every input with probability $1 - \delta$

Complexity Measures

Functions of the input size:

- Information Cost: $\mathrm{ICost}_\mu(\Pi) = I(X : \Pi(X))$
  where
  - $X =$ input selected from μ
  - $\Pi(X)$ transcript of Π on X
- Information Complexity: $\mathrm{IC}_{\mu,\delta}(P) = \inf_\Pi \mathrm{ICost}_\mu(\Pi)$
  where
  - infimum is taken over protocols Π that solve problem P with probability $1-\delta$
- Communication Complexity: $\mathrm{CC}_\delta(P) = \inf_\Pi \max_X |\Pi(X)|$
  where
  - infimum is taken over protocols Π that solve problem P on every input with probability $1-\delta$

Easy to prove: $\mathrm{CC}_\delta(P) \geq \mathrm{IC}_{\mu,\delta}(P)$

Proof Overview

Problem BBB (Basic Building Block):

- $2p$ players with two instances of pointer chasing

Proof Overview

Problem BBB (Basic Building Block):

- $2p$ players with two instances of pointer chasing
- Problem to solve: Is the result the same?

Proof Overview

Problem BBB (Basic Building Block):

- $2p$ players with two instances of pointer chasing
- Problem to solve: Is the result the same?
- (Actually: If some function maps $\Omega(\log n)$ elements to one element, also say YES)

Proof Overview

Problem BBB (Basic Building Block):

- $2p$ players with two instances of pointer chasing
- Problem to solve: Is the result the same?
- (Actually: If some function maps $\Omega(\log n)$ elements to one element, also say YES)

Three Steps:

1. $\mathrm{IC}_{\mu,1/n^2}(\text{BBB}) \approx \Omega(n)$

Proof Overview

Problem BBB (Basic Building Block):

- $2p$ players with two instances of pointer chasing
- Problem to solve: Is the result the same?
- (Actually: If some function maps $\Omega(\log n)$ elements to one element, also say YES)

Three Steps:

1. $\mathrm{IC}_{\mu,1/n^2}(\text{BBB}) \approx \Omega(n)$
2. $\mathrm{IC}_{\mu^k,1/(2n^2)}(\bigvee_{i=1}^{k} \text{BBB}) \approx k \cdot \mathrm{IC}_{\mu,1/n^2}(\text{BBB}) \approx \Omega(kn)$ for $k \ll n$

Proof Overview

Problem BBB (Basic Building Block):

- $2p$ players with two instances of pointer chasing
- Problem to solve: Is the result the same?
- (Actually: If some function maps $\Omega(\log n)$ elements to one element, also say YES)

Three Steps:

1. $\mathrm{IC}_{\mu,1/n^2}(\text{BBB}) \approx \Omega(n)$
2. $\mathrm{IC}_{\mu^k,1/(2n^2)}(\bigvee_{i=1}^{k} \text{BBB}) \approx k \cdot \mathrm{IC}_{\mu,1/n^2}(\text{BBB}) \approx \Omega(kn)$ for $k \ll n$
   Implies: $\mathrm{CC}_{1/10}(\bigvee_{i=1}^{k} \text{BBB}) \gtrsim \Omega(kn)$

Proof Overview

Problem BBB (Basic Building Block):

- $2p$ players with two instances of pointer chasing
- Problem to solve: Is the result the same?
- (Actually: If some function maps $\Omega(\log n)$ elements to one element, also say YES)

Three Steps:

1. $\mathrm{IC}_{\mu,1/n^2}(\text{BBB}) \approx \Omega(n)$
2. $\mathrm{IC}_{\mu^k,1/(2n^2)}(\bigvee_{i=1}^{k} \text{BBB}) \approx k \cdot \mathrm{IC}_{\mu,1/n^2}(\text{BBB}) \approx \Omega(kn)$ for $k \ll n$
   Implies: $\mathrm{CC}_{1/10}(\bigvee_{i=1}^{k} \text{BBB}) \gtrsim \Omega(kn)$
3. $\mathrm{CC}_{1/20}(\text{BFS tree intersection}) \gtrsim \mathrm{CC}_{1/10}(\bigvee_{i=1}^{k} \text{BBB})$
   for $k = n^{\Theta(1/p)}$

Step 1

Statement:

$$\mathrm{IC}_{\mu,1/n^2}(\mathsf{BBB}) \approx \Omega(n)$$

Step 1

Statement:

$$\mathrm{IC}_{\mu,1/n^2}(\mathsf{BBB}) \approx \Omega(n)$$

How (1/2):

- Modify [Nisan, Wigderson 1993] or [Guha, McGregor 2007]:

Step 1

Statement:

$$\mathrm{IC}_{\mu,1/n^2}(\mathsf{BBB}) \approx \Omega(n)$$

How (1/2):

- Modify [Nisan, Wigderson 1993] or [Guha, McGregor 2007]:
    - Look at the leaves in the protocol tree

Step 1

Statement:

$$\mathrm{IC}_{\mu,1/n^2}(\textsf{BBB}) \approx \Omega(n)$$

How (1/2):

- Modify [Nisan, Wigderson 1993] or [Guha, McGregor 2007]:
  - Look at the leaves in the protocol tree
  - They show: high entropy of $f_k(\ldots)$

Step 1

Statement:

$$\mathrm{IC}_{\mu,1/n^2}(\mathsf{BBB}) \approx \Omega(n)$$

How (1/2):

- Modify [Nisan, Wigderson 1993] or [Guha, McGregor 2007]:
  - Look at the leaves in the protocol tree
  - They show: high entropy of $f_k(\ldots)$
  - We show: high entropy of both $f_k(\ldots)$ and $g_k(\ldots)$

Step 1

Statement:

$$\mathrm{IC}_{\mu,1/n^2}(\mathsf{BBB}) \approx \Omega(n)$$

How (1/2):

- Modify [Nisan, Wigderson 1993] or [Guha, McGregor 2007]:
  - Look at the leaves in the protocol tree
  - They show: high entropy of $f_k(\ldots)$
  - We show: high entropy of both $f_k(\ldots)$ and $g_k(\ldots)$
    $\Rightarrow$ w.p. $\Omega(1/n)$, $f_k(\ldots) = g_k(\ldots)$ and protocol incorrect

Step 1

Statement:

$$\mathrm{IC}_{\mu,1/n^2}(\textsf{BBB}) \approx \Omega(n)$$

How (1/2):

- Modify [Nisan, Wigderson 1993] or [Guha, McGregor 2007]:
  - Look at the leaves in the protocol tree
  - They show: high entropy of $f_k(\dots)$
  - We show: high entropy of both $f_k(\dots)$ and $g_k(\dots)$ $\Rightarrow$ w.p. $\Omega(1/n)$, $f_k(\dots) = g_k(\dots)$ and protocol incorrect
  - Obtain lower bound for communication complexity on random input

Step 1

Statement:

$$\mathrm{IC}_{\mu,1/n^2}(\mathsf{BBB}) \approx \Omega(n)$$

How (2/2):

- Use [Jain, Radhakrishnan, Sen 2003]?

Step 1

Statement:

$$\mathrm{IC}_{\mu,1/n^2}(\textsf{BBB}) \approx \Omega(n)$$

How (2/2):

- Use [Jain, Radhakrishnan, Sen 2003]?
  $\Pi =$ protocol w/small $\mathrm{ICost}_\mu(\Pi)$, $\Theta(1)$ messages, ϵ error

Step 1

Statement:

$$\mathrm{IC}_{\mu,1/n^2}(\textsf{BBB}) \approx \Omega(n)$$

How (2/2):

- Use [Jain, Radhakrishnan, Sen 2003]?
  $\Pi =$ protocol w/small $\mathrm{ICost}_\mu(\Pi)$, $\Theta(1)$ messages, ϵ error

  There is protocol Π' with total communication $\sim \mathrm{ICost}_\mu(\Pi)/\delta^2$ that errs with probability $\epsilon + \delta$

Step 1

Statement:

$$\mathrm{IC}_{\mu,1/n^2}(\textsf{BBB}) \approx \Omega(n)$$

How (2/2):

- Use [Jain, Radhakrishnan, Sen 2003]?
  $\Pi =$ protocol w/small $\mathrm{ICost}_\mu(\Pi)$, $\Theta(1)$ messages, ϵ error

  There is protocol Π' with total communication $\sim \mathrm{ICost}_\mu(\Pi)/\delta^2$ that errs with probability $\epsilon + \delta$

- Won't work! $\delta = o(1/n)$

Step 1

Statement:

$$\mathrm{IC}_{\mu,1/n^2}(\text{BBB}) \approx \Omega(n)$$

How (2/2):

- Use [Jain, Radhakrishnan, Sen 2003]?
  $\Pi =$ protocol w/small $\mathrm{ICost}_\mu(\Pi)$, $\Theta(1)$ messages, ϵ error

  There is protocol Π' with total communication $\sim \mathrm{ICost}_\mu(\Pi)/\delta^2$ that errs with probability $\epsilon + \delta$

- Won't work! $\delta = o(1/n)$
- Resolution:
  - Modify the [JRS] technique to make Π' send short messages most of the time

Step 1

Statement:

$$\mathrm{IC}_{\mu,1/n^2}(\text{BBB}) \approx \Omega(n)$$

How (2/2):

- Use [Jain, Radhakrishnan, Sen 2003]?
  $\Pi =$ protocol w/small $\mathrm{ICost}_\mu(\Pi)$, $\Theta(1)$ messages, ϵ error

  There is protocol Π' with total communication $\sim \mathrm{ICost}_\mu(\Pi)/\delta^2$ that errs with probability $\epsilon + \delta$

- Won't work! $\delta = o(1/n)$
- Resolution:
  - Modify the [JRS] technique to make Π' send short messages most of the time
  - Modify [Nisan, Wigderson 1993] to work on such protocols

Step 2

Statement:

$\mathrm{IC}_{\mu^k,1/(2n^2)}(\bigvee_{i=1}^{k} \textsf{BBB}) \approx k \cdot \mathrm{IC}_{\mu,1/n^2}(\textsf{BBB}) \approx \Omega(kn)$ for $k \ll n$

Step 2

Statement:

$\mathrm{IC}_{\mu^k,1/(2n^2)}(\bigvee_{i=1}^k \textsf{BBB}) \approx k \cdot \mathrm{IC}_{\mu,1/n^2}(\textsf{BBB}) \approx \Omega(kn)$ for $k \ll n$

How:

- Product distribution:
  information cost $= \sum_{i=1}^k$ information cost for instance i

Step 2

Statement:

$\mathrm{IC}_{\mu^k,1/(2n^2)}(\bigvee_{i=1}^{k} \textsf{BBB}) \approx k \cdot \mathrm{IC}_{\mu,1/n^2}(\textsf{BBB}) \approx \Omega(kn)$ for $k \ll n$

How:

- Product distribution:
  information cost $= \sum_{i=1}^{k}$ information cost for instance i
- For specific instance, $\bigvee$(other instances) $=$ false most of the time

Step 2

Statement:

$\mathrm{IC}_{\mu^k,1/(2n^2)}(\bigvee_{i=1}^{k} \textsf{BBB}) \approx k \cdot \mathrm{IC}_{\mu,1/n^2}(\textsf{BBB}) \approx \Omega(kn)$ for $k \ll n$

How:

- Product distribution:
  information cost $= \sum_{i=1}^{k}$ information cost for instance i
- For specific instance, $\bigvee$(other instances) $=$ false most of the time
- Information cost won't decrease significantly on $\bigvee$(other instances) $=$ true

Step 3

Statement:

$$\mathrm{CC}_{1/20}(\text{BFS tree intersection}) \gtrsim \mathrm{CC}_{1/10}(\bigvee_{i=1}^{k} \text{BBB})$$

for $k = n^{\Theta(1/p)}$

Step 3

Statement:

$$\mathrm{CC}_{1/20}(\text{BFS tree intersection}) \gtrsim \mathrm{CC}_{1/10}(\bigvee_{i=1}^{k} \text{BBB})$$

for $k = n^{\Theta(1/p)}$

How:

- Want: Show protocol for $\bigvee_{i=1}^{k}$ BBB

Step 3

Statement:

$$\mathrm{CC}_{1/20}(\text{BFS tree intersection}) \gtrsim \mathrm{CC}_{1/10}(\bigvee_{i=1}^{k} \text{BBB})$$

for $k = n^{\Theta(1/p)}$

How:

- Want: Show protocol for $\bigvee_{i=1}^{k}$ BBB
- Randomly relabel intermediate results of functions and stack them on top of each other
  $\Rightarrow$ instance for (BFS tree intersection)

Step 3

Statement:

$$\mathrm{CC}_{1/20}(\text{BFS tree intersection}) \gtrsim \mathrm{CC}_{1/10}(\bigvee_{i=1}^{k} \text{BBB})$$

for $k = n^{\Theta(1/p)}$

How:

- Want: Show protocol for $\bigvee_{i=1}^{k}$ BBB
- Randomly relabel intermediate results of functions and stack them on top of each other
  $\Rightarrow$ instance for (BFS tree intersection)
- Result:
  - If pair of pointer chasing instances gave the same element, BFS trees will intersection

Step 3

Statement:

$$\mathrm{CC}_{1/20}(\text{BFS tree intersection}) \gtrsim \mathrm{CC}_{1/10}(\bigvee_{i=1}^{k} \text{BBB})$$

for $k = n^{\Theta(1/p)}$

How:

- Want: Show protocol for $\bigvee_{i=1}^{k}$ BBB
- Randomly relabel intermediate results of functions and stack them on top of each other
  $\Rightarrow$ instance for (BFS tree intersection)
- Result:
  - If pair of pointer chasing instances gave the same element, BFS trees will intersection
  - If no pair gave the same element and no $\Theta(\log n)$-to-1 mapping, BFS trees unlikely to intersect

Step 3

Statement:

$$\mathrm{CC}_{1/20}(\text{BFS tree intersection}) \gtrsim \mathrm{CC}_{1/10}(\bigvee_{i=1}^{k} \text{BBB})$$

for $k = n^{\Theta(1/p)}$

How:

- Want: Show protocol for $\bigvee_{i=1}^{k}$ BBB
- Randomly relabel intermediate results of functions and stack them on top of each other
  $\Rightarrow$ instance for (BFS tree intersection)
- Result:
  - If pair of pointer chasing instances gave the same element, BFS trees will intersection
  - If no pair gave the same element and no $\Theta(\log n)$-to-1 mapping, BFS trees unlikely to intersect
  - $\Theta(2p)$ additional communication to notice $\Theta(\log n)$-to-1 mapping $\Rightarrow$ output “YES”

What we prove:

Problems “Is there a perfect matching?”
and “Are v and w at distance $2p$?”
in $< p$ passes require $\sim n^{1+\Omega(1/p)}$ space

What we prove:

Problems "Is there a perfect matching?"
and "Are v and w at distance $2p$?"
in $< p$ passes require $\sim n^{1+\Omega(1/p)}$ space

Open Questions:

- Can simplify?

What we prove:

Problems “Is there a perfect matching?”
and “Are v and w at distance $2p$?”
in $< p$ passes require $\sim n^{1+\Omega(1/p)}$ space

Open Questions:

- Can simplify?
- Better bound: Is looking for a few augmenting paths harder?

What we prove:

Problems “Is there a perfect matching?”
and “Are v and w at distance $2p$?”
in $< p$ passes require $\sim n^{1+\Omega(1/p)}$ space

Open Questions:

- Can simplify?
- Better bound: Is looking for a few augmenting paths harder?
- Can the techniques be used for approximate matchings?

Questions?