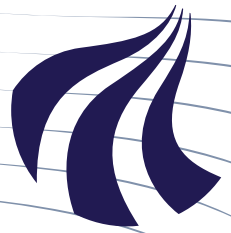
Self-Adjusting Linear Networks

with Chen Avin* and Stefan Schmid†

*Ben Gurion University, †University of Vienna

Ingo van Duijn

Department of Computer Science
Aalborg University
Denmark



AALBORG UNIVERSITY
DENMARK

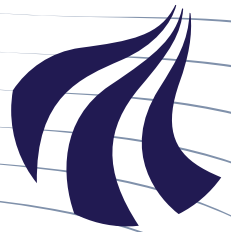
Baby Steps towards Self-Adjusting Linear Networks

with Chen Avin* and Stefan Schmid†

*Ben Gurion University, †University of Vienna

Ingo van Duijn

Department of Computer Science
Aalborg University
Denmark



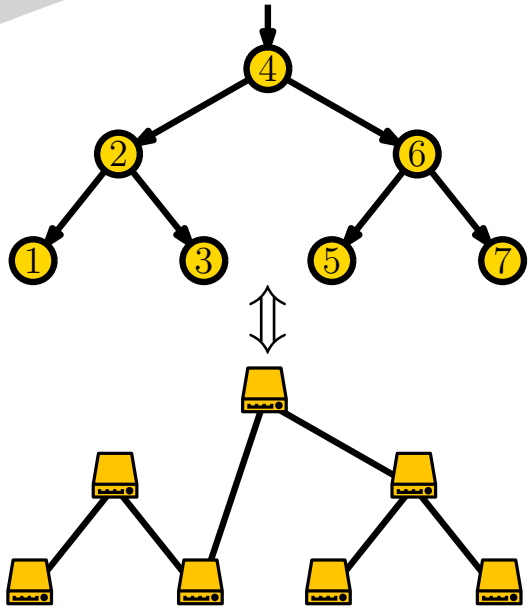
AALBORG UNIVERSITY
DENMARK

Overview

Overview

Intro: correspondence between

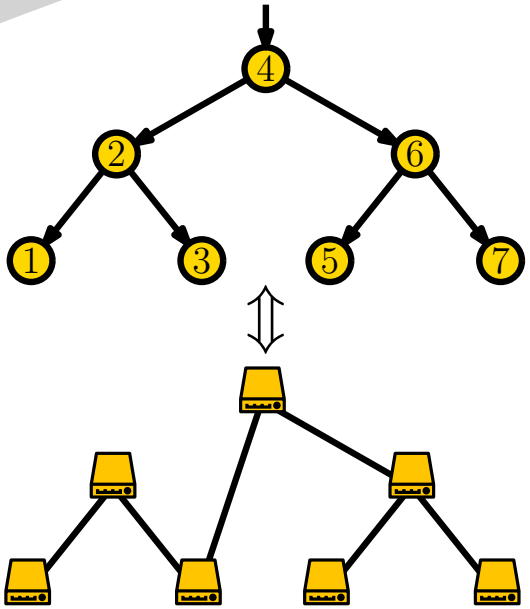
- Self-Adjusting Networks
- Dynamic Data structures

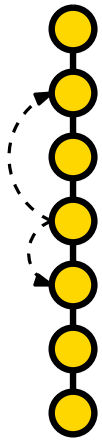


Overview

Intro: correspondence between
- Self-Adjusting Networks
- Dynamic Data structures

Case study:
- List Access
- List Communication



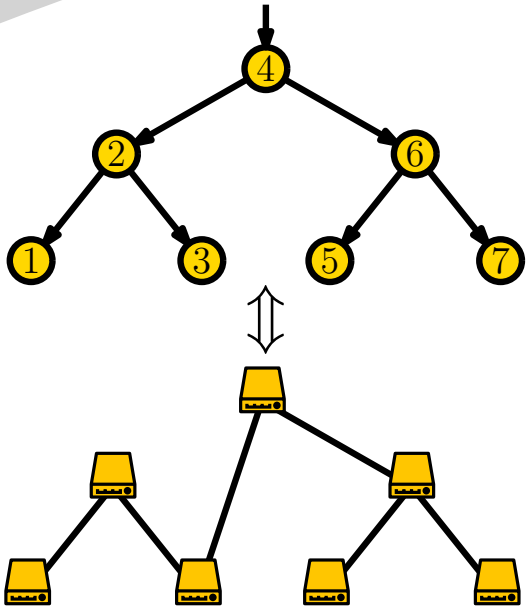


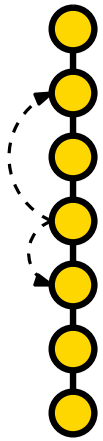
Overview

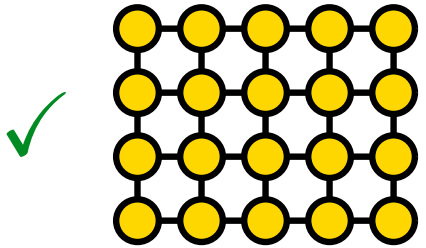
Intro: correspondence between
- Self-Adjusting Networks
- Dynamic Data structures

Case study:
- List Access
- List Communication

Results:
- Grid Network Bounds







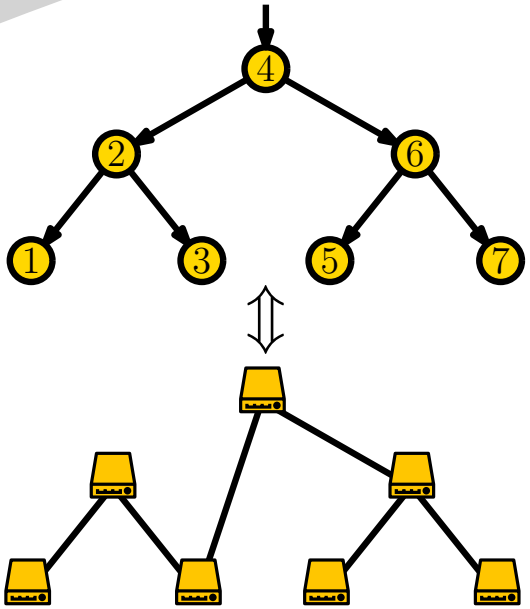
Overview

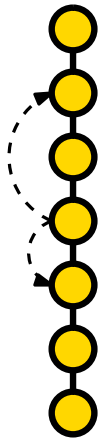
Intro: correspondence between
- Self-Adjusting Networks
- Dynamic Data structures

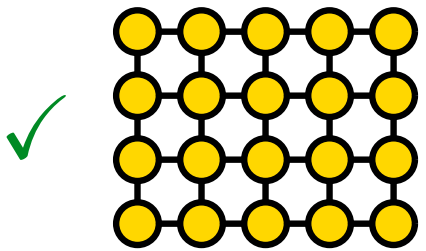
Case study:
- List Access
- List Communication

Results:
- Grid Network Bounds

Lower Bound Proof Sketch







Overview

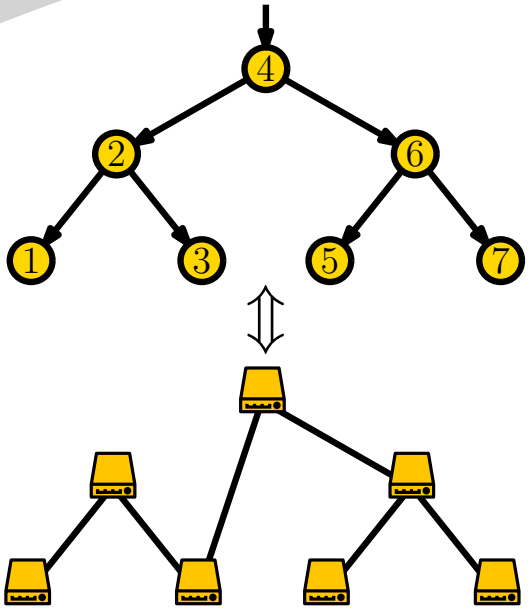
Intro: correspondence between
- Self-Adjusting Networks
- Dynamic Data structures

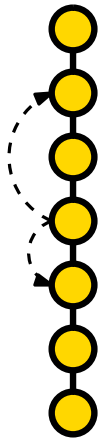
Case study:
- List Access
- List Communication

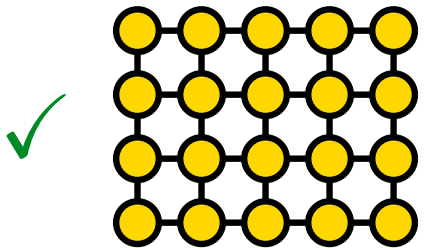
Results:
- Grid Network Bounds

Lower Bound Proof Sketch

Future Work





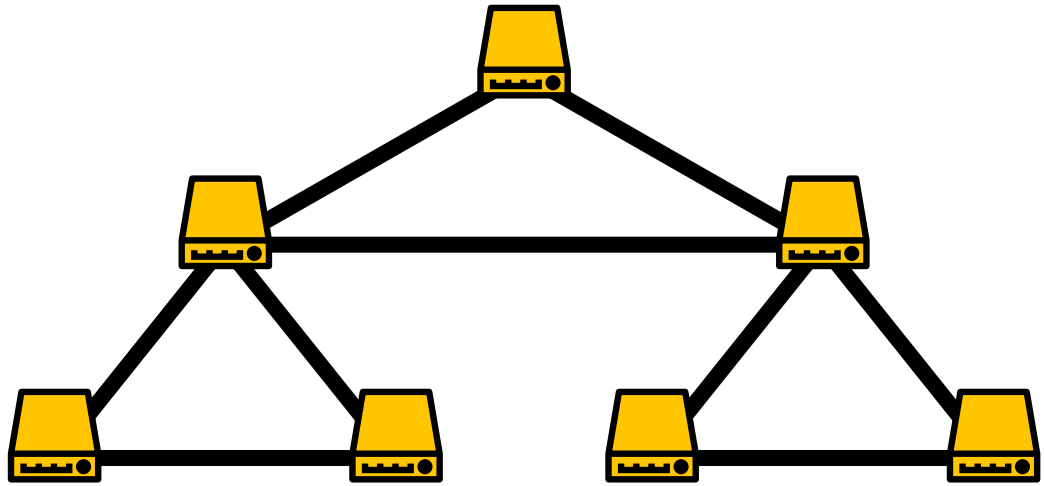


Self-Adjusting Networks

Self-Adjusting Networks

The Setting:

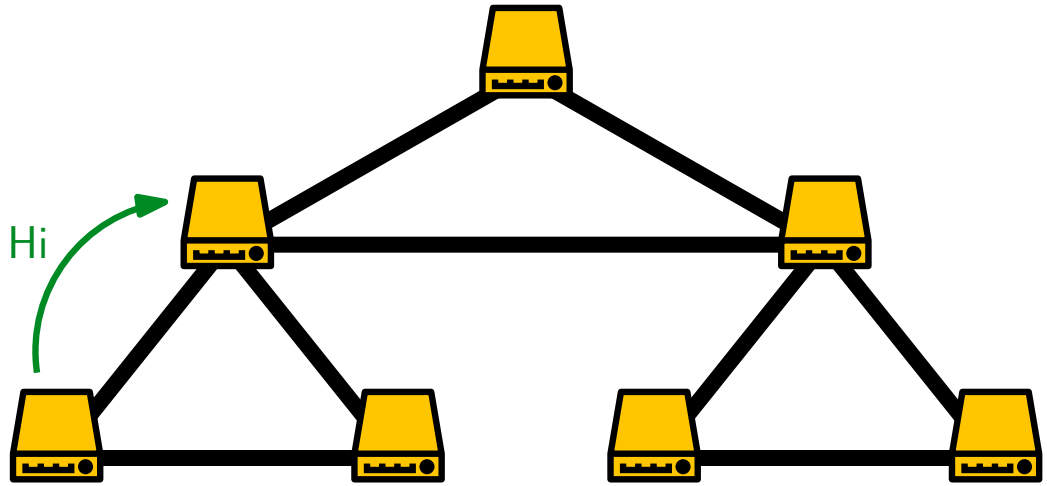
- Network



Self-Adjusting Networks

The Setting:

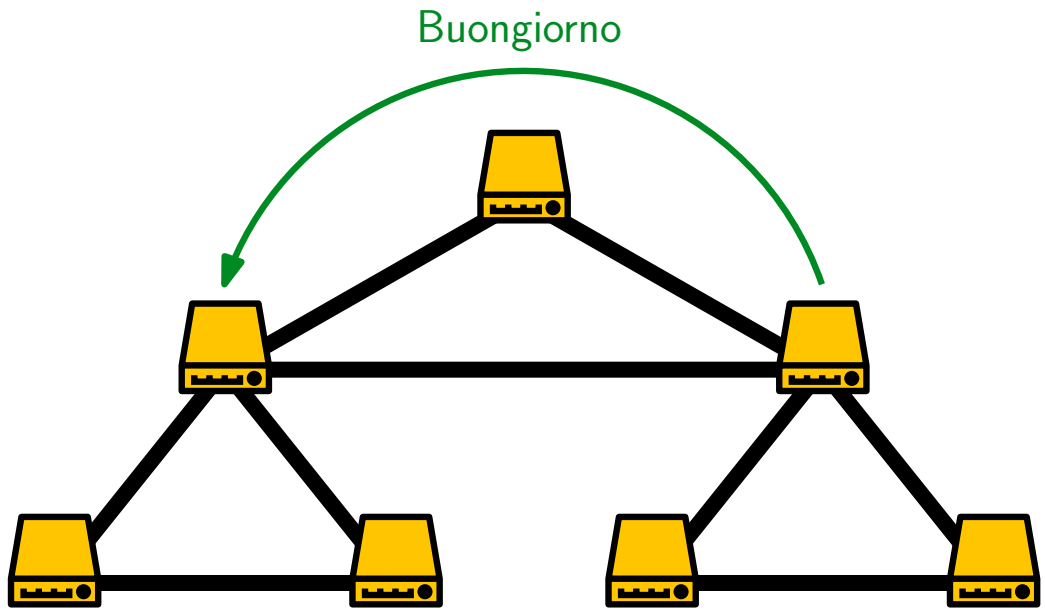
- Network
- Communication



Self-Adjusting Networks

The Setting:

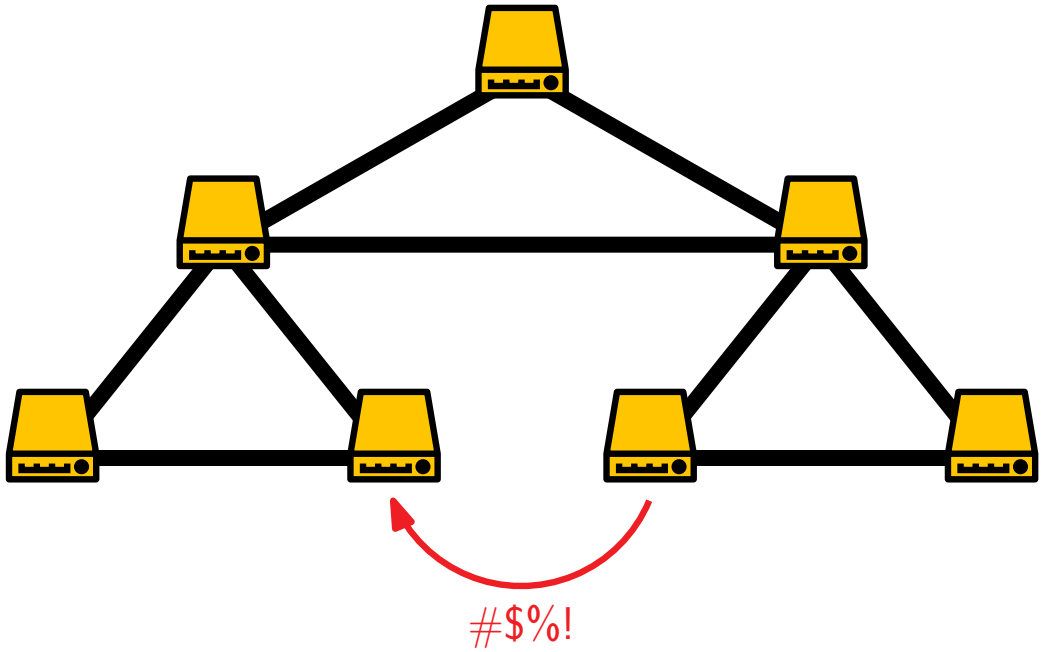
- Network
- Communication



Self-Adjusting Networks

The Setting:

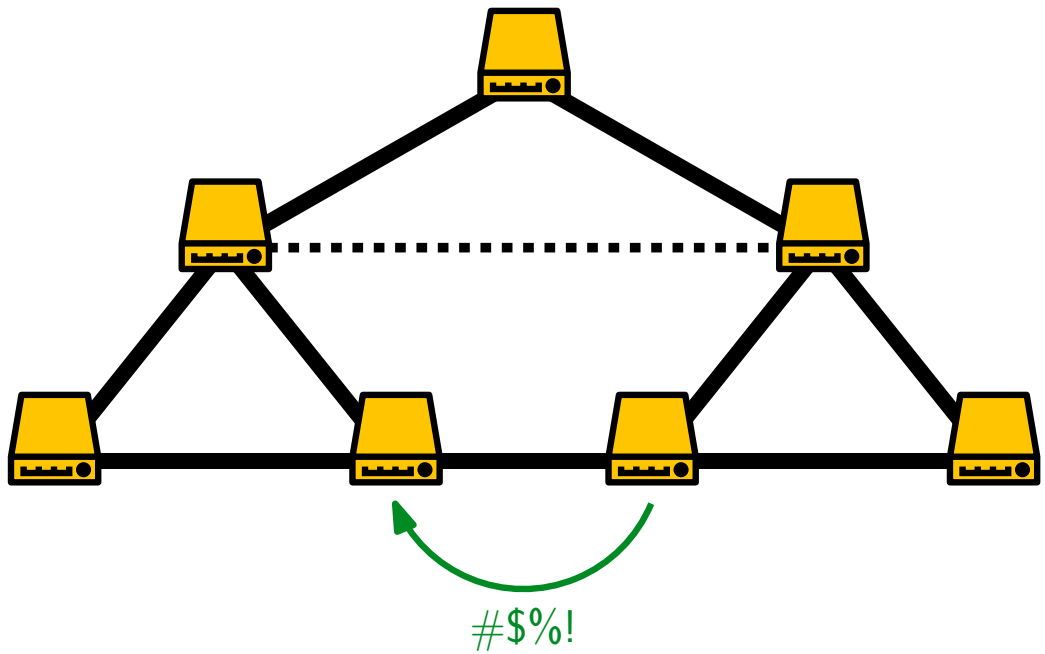
- Network
- Communication



Self-Adjusting Networks

The Setting:

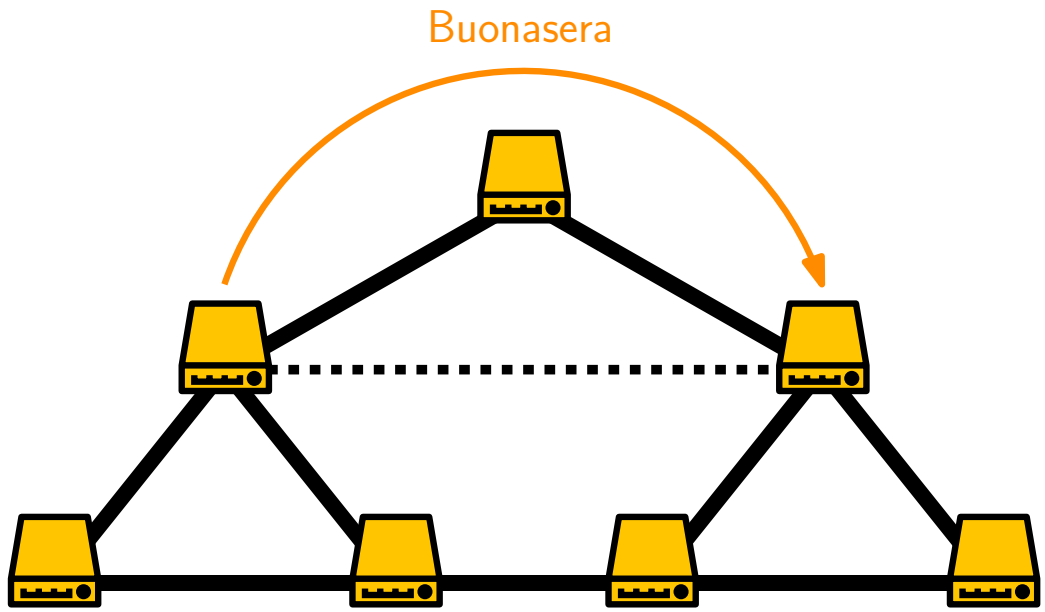
- Network
- Communication
- Adjustments



Self-Adjusting Networks

The Setting:

- Network
- Communication
- Adjustments
- Online



Self-Adjusting Networks

The Setting:

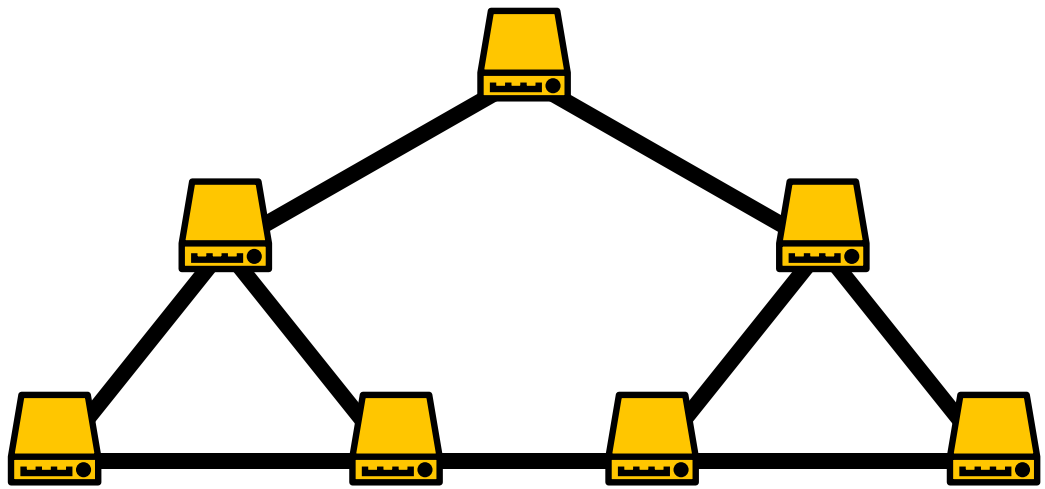
- Network
- Communication
- Adjustments

} Cost!

- Online

The Goal:

- Minimise Cost



Self-Adjusting Networks

The Setting:

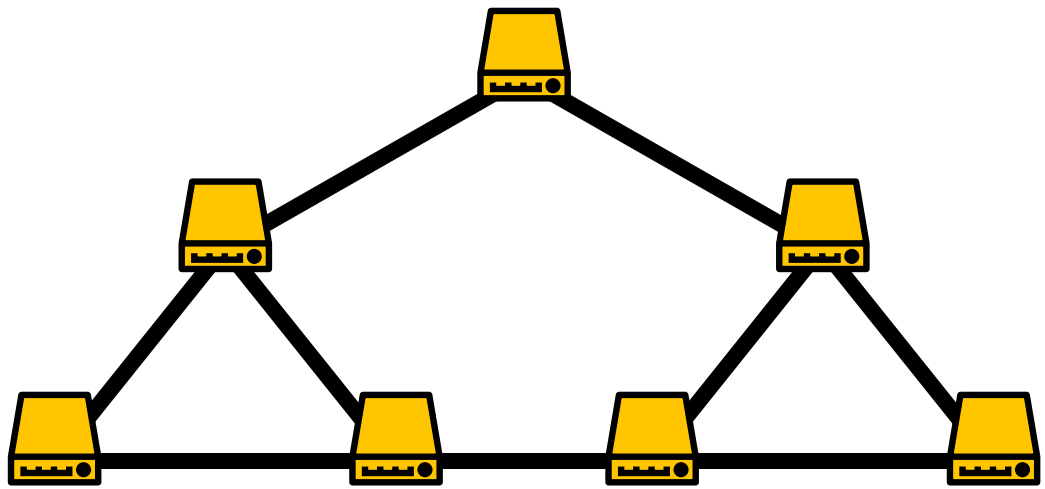
- Network
- Communication
- Adjustments

} Cost!

- Online

The Goal:

- Minimise Cost
- (Distributed Algorithm)



Self-Adjusting Networks

The Setting:

- Network
- Communication
- Adjustments

} Cost!

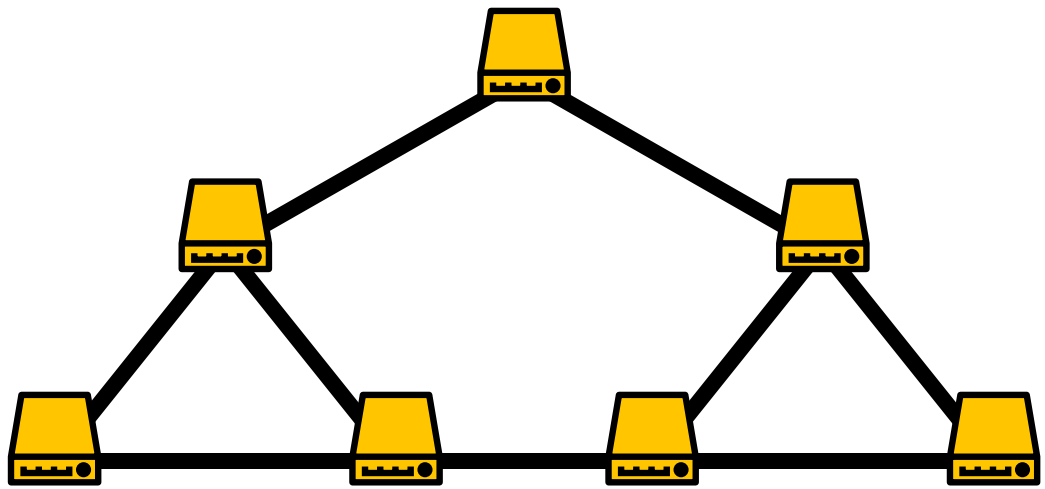
- Online

The Goal:

- Minimise Cost
- (Distributed Algorithm)

The Questions:

- Which Model?



Self-Adjusting Networks

The Setting:

- Network
- Communication
- Adjustments
- Online

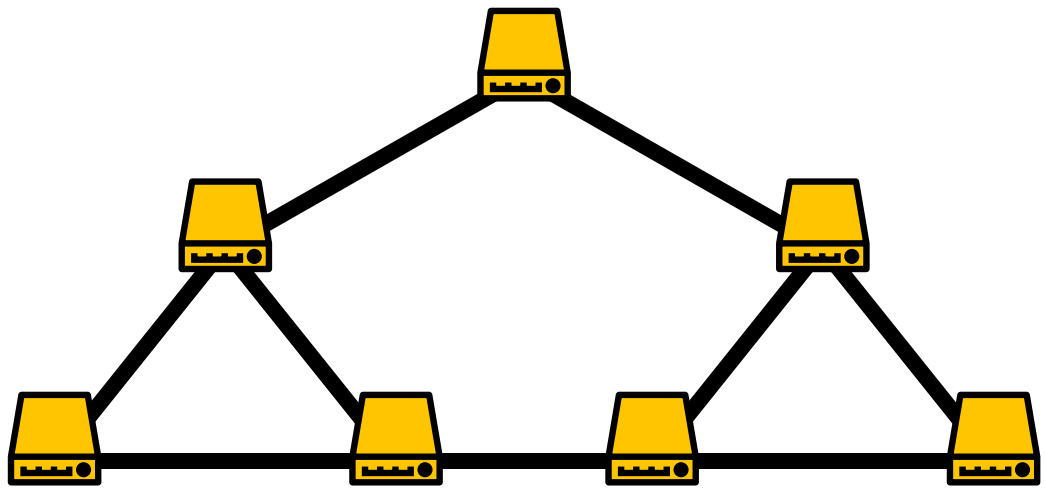
} Cost!

The Goal:

- Minimise Cost
- (Distributed Algorithm)

The Questions:

- Which Model?
- Formal Guarantees?

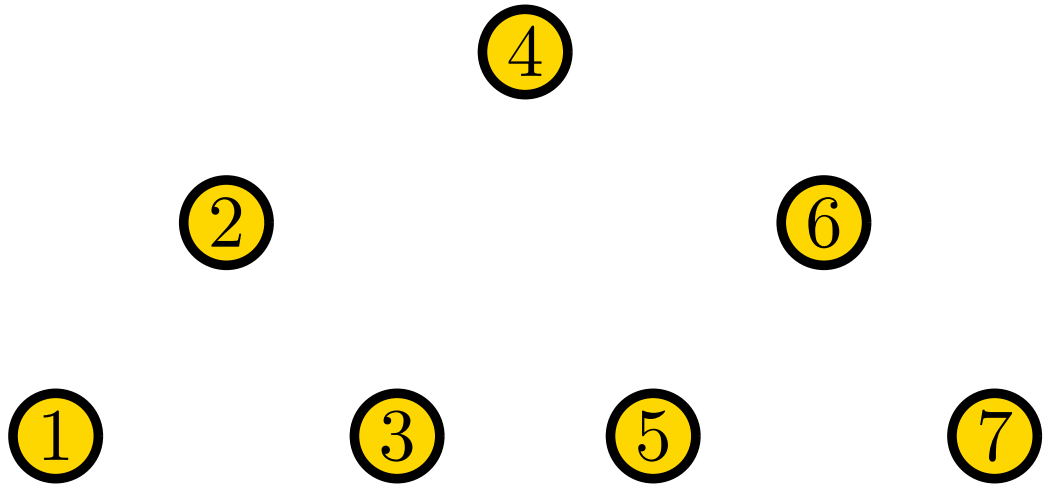


Dynamic Dictionaries

Dynamic Dictionaries

The Setting:

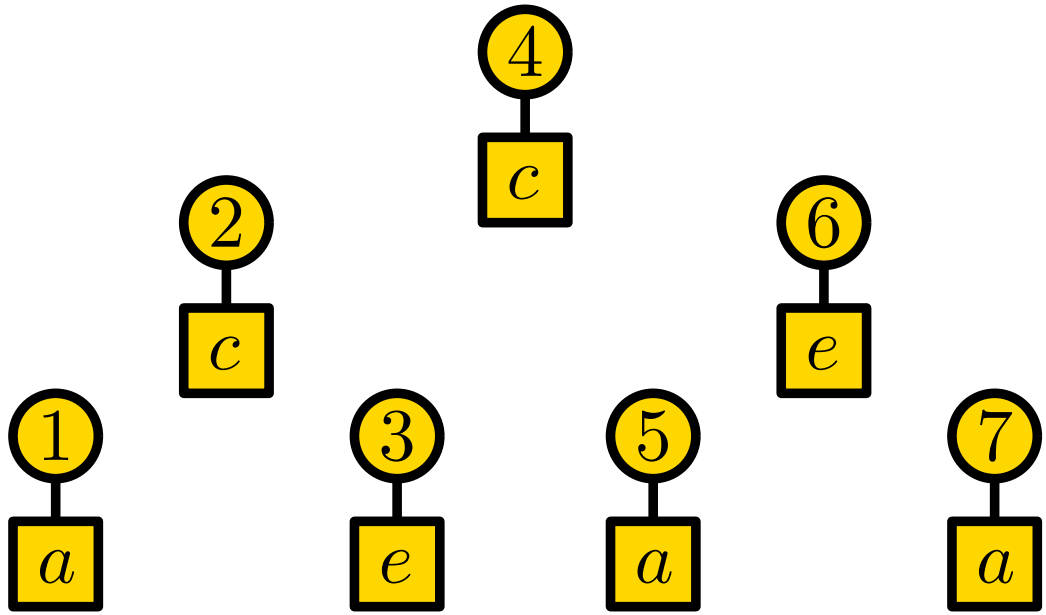
- Keys



Dynamic Dictionaries

The Setting:

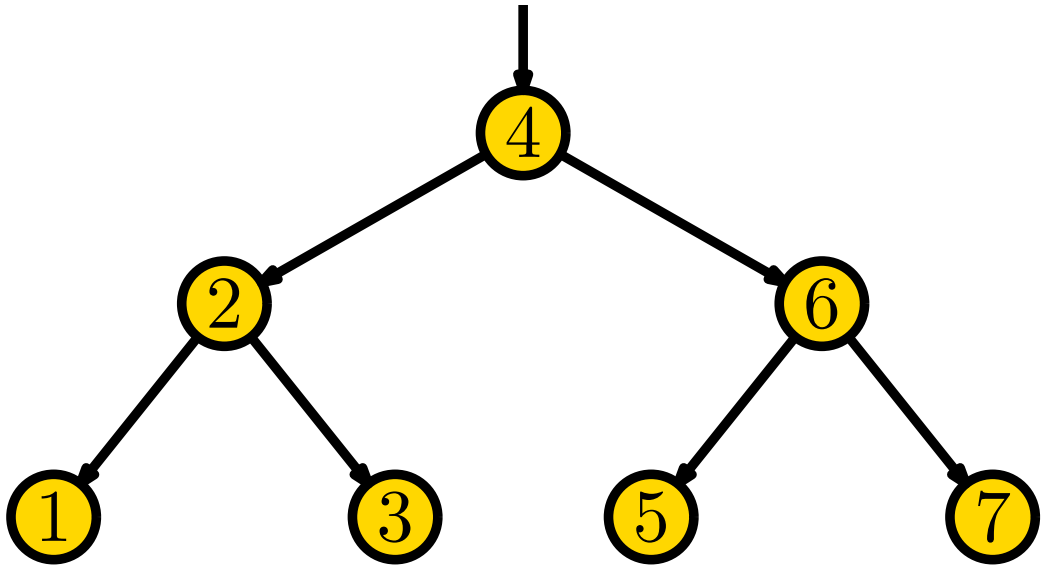
- Keys (with data)



Dynamic Dictionaries

The Setting:

- Keys
- Pointer Structure

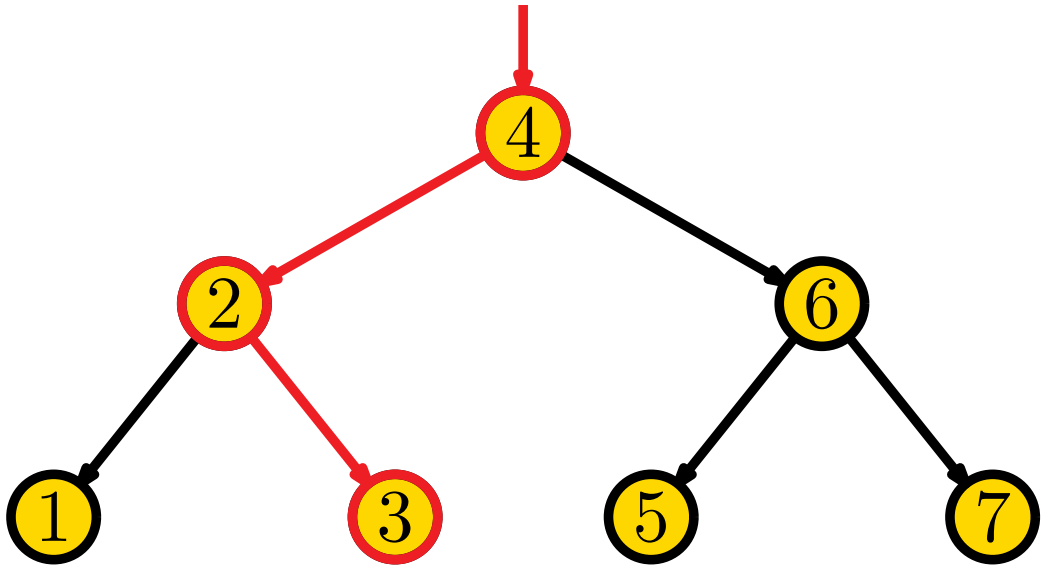


Dynamic Dictionaries

The Setting:

- Keys
- Pointer Structure
- Key Access

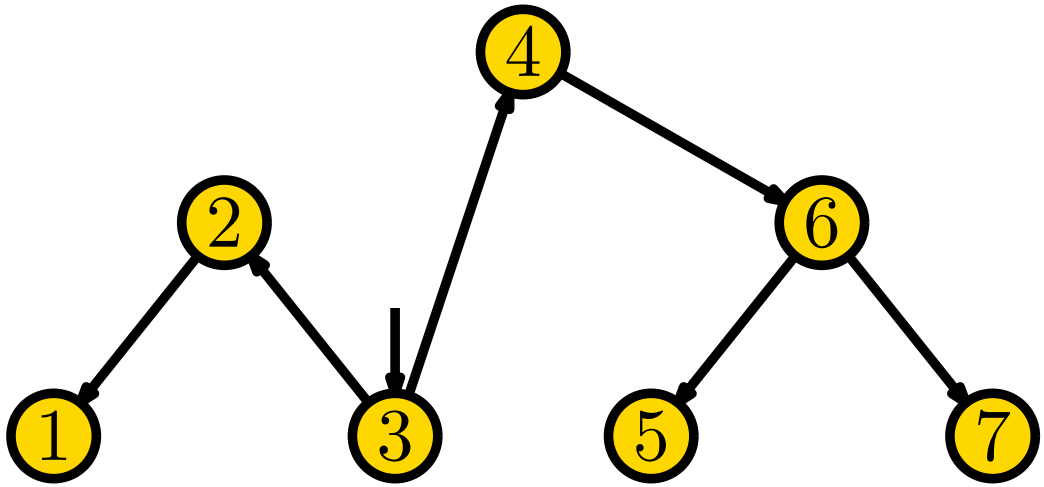
Access 3



Dynamic Dictionaries

The Setting:

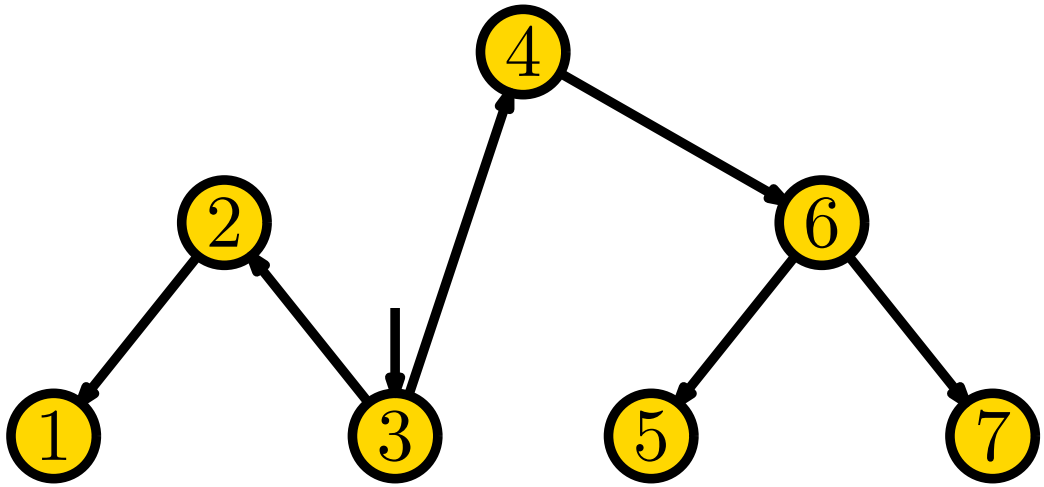
- Keys
- Pointer Structure
- Key Access
- Adjustments



Dynamic Dictionaries

The Setting:

- Keys
- Pointer Structure
- Key Access
- Adjustments
- Online



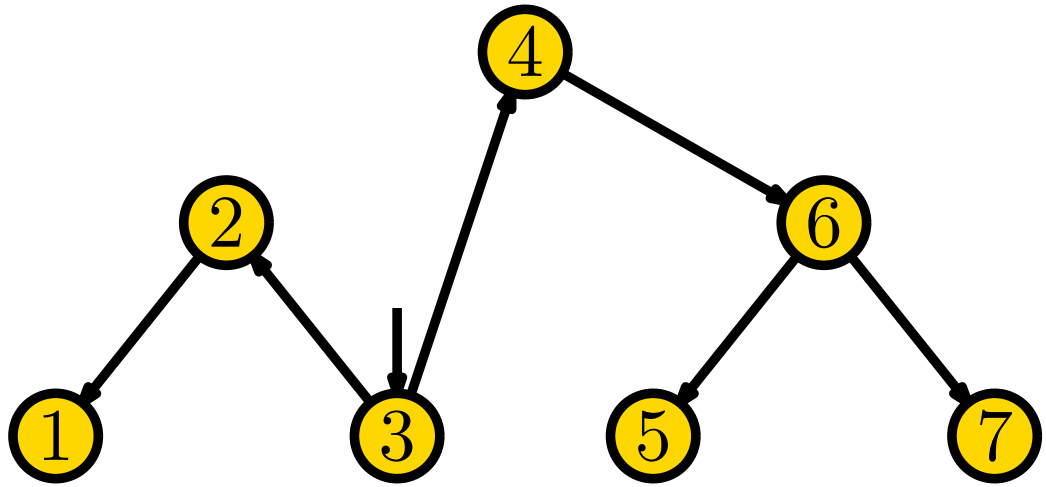
Dynamic Dictionaries

The Setting:

- Keys
- Pointer Structure
- Key Access
- Adjustments
- Online

The Goal:

- Dynamic Optimalty



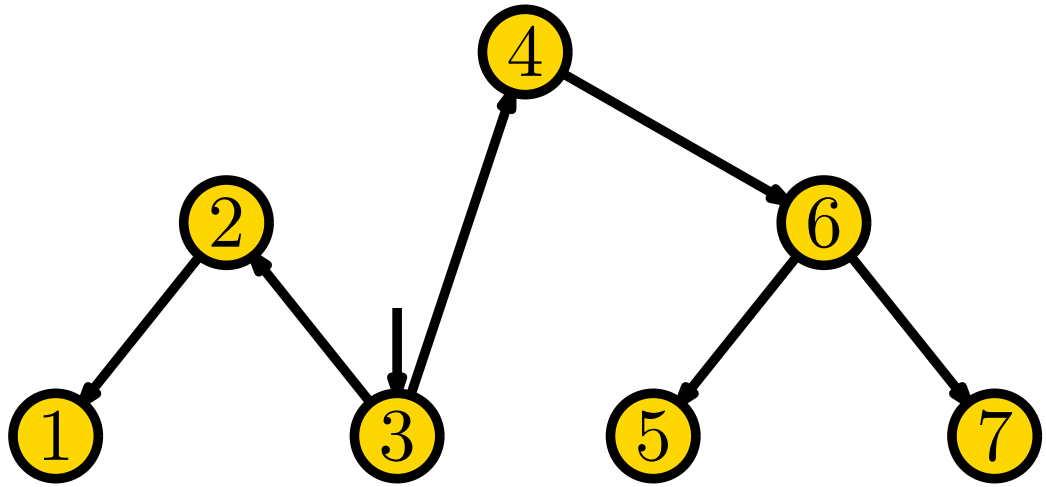
Dynamic Dictionaries

The Setting:

- Keys
- Pointer Structure
- Key Access
- Adjustments
- Online

The Goal:

- Dynamic Optimalty
- Minimise: $\frac{\text{Best Online Cost}}{\text{Best Offline Cost}}$



Dynamic Dictionaries

The Setting:

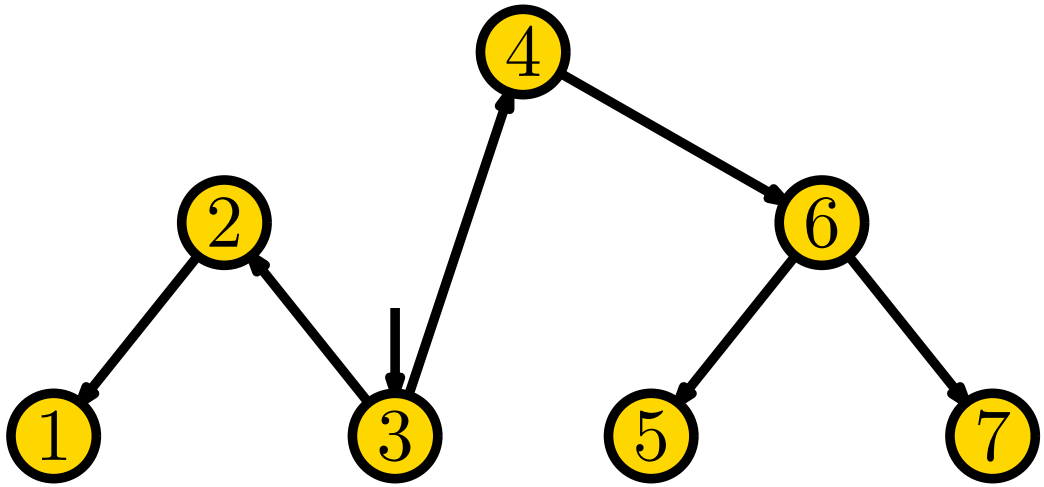
- Keys
- Pointer Structure
- Key Access
- Adjustments
- Online

The Goal:

- Dynamic Optimalty
- Minimise: $\frac{\text{Best Online Cost}}{\text{Best Offline Cost}}$

Towards Networks:

- Pairwise Access



Dynamic Dictionaries

The Setting:

- Keys
- Pointer Structure
- Key Access
- Adjustments
- Online

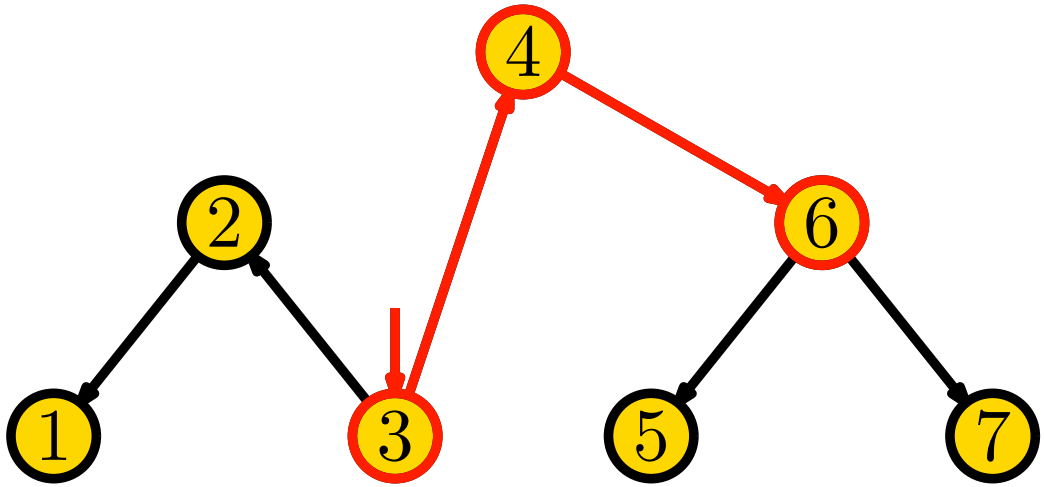
The Goal:

- Dynamic Optimalty
- Minimise: $\frac{\text{Best Online Cost}}{\text{Best Offline Cost}}$

Towards Networks:

- Pairwise Access

Access 6



Dynamic Dictionaries

The Setting:

- Keys
- Pointer Structure
- Key Access
- Adjustments
- Online

The Goal:

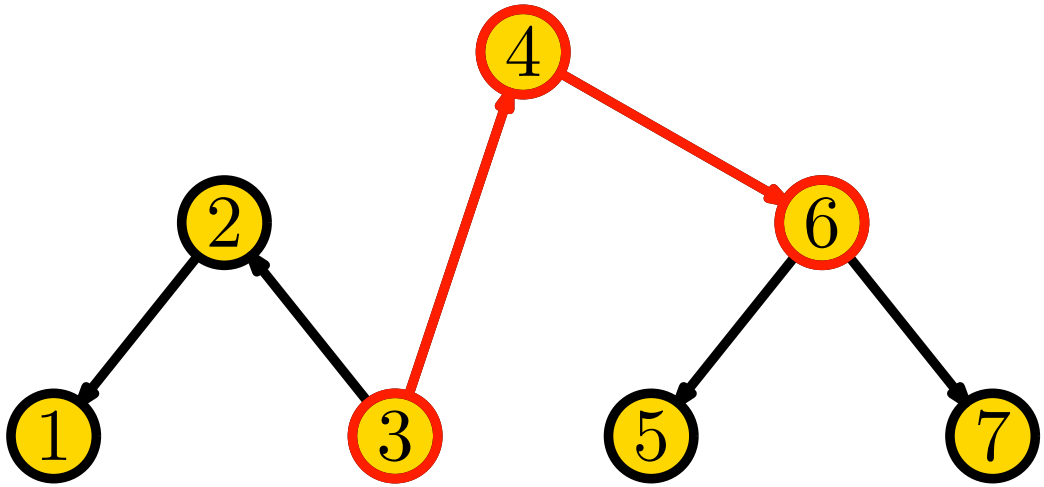
- Dynamic Optimalty
- Minimise: $\frac{\text{Best Online Cost}}{\text{Best Offline Cost}}$

Towards Networks:

- Pairwise Access

~~Access 6~~

Communicate $(3, 6)$



Dynamic Dictionaries

The Setting:

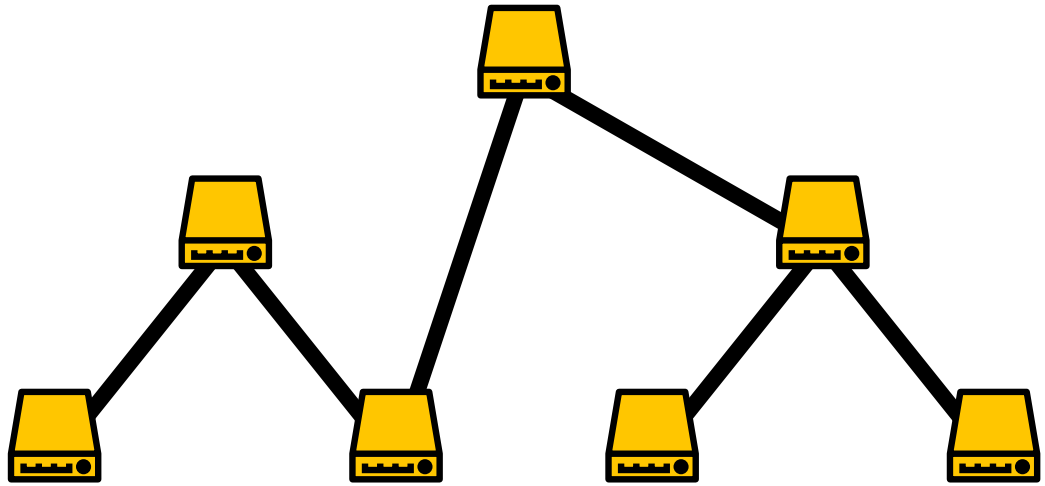
- Keys
- Pointer Structure
- Key Access
- Adjustments
- Online

The Goal:

- Dynamic Optimalty
- Minimise: $\frac{\text{Best Online Cost}}{\text{Best Offline Cost}}$

Towards Networks:

- Pairwise Access
- Ignore Searching/Routing

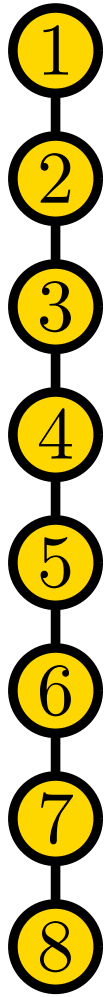


A Simple$^{?}$ Network

A Simple[?] Network

The Dictionary:

- Linked List



A Simple[?] Network

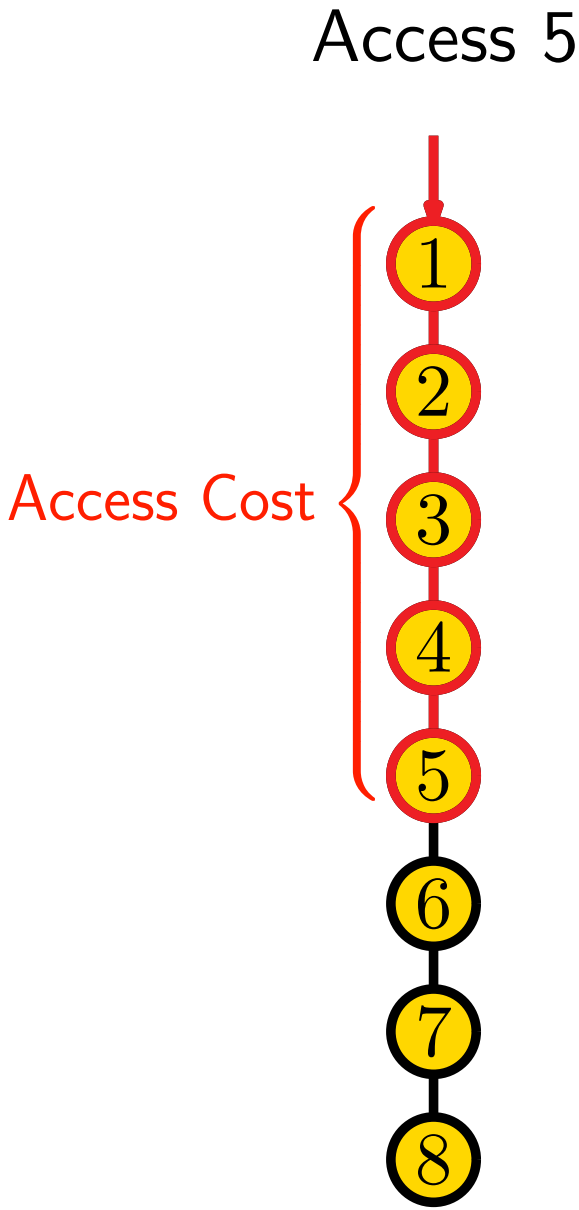
The Dictionary:

- Linked List
- Access in Front

A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front



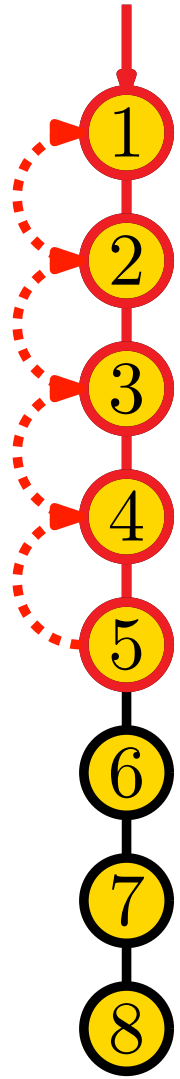
A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm

Access 5

Move Cost



A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm

Access 5

A Simple$^{?}$ Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  $\rightarrow$ Dynamically Optimal!

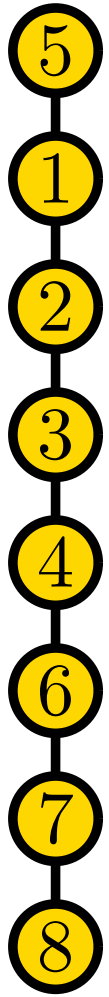
A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  $\rightarrow$ Dynamically Optimal!

The Network:

- Pairwise Access



A Simple[?] Network

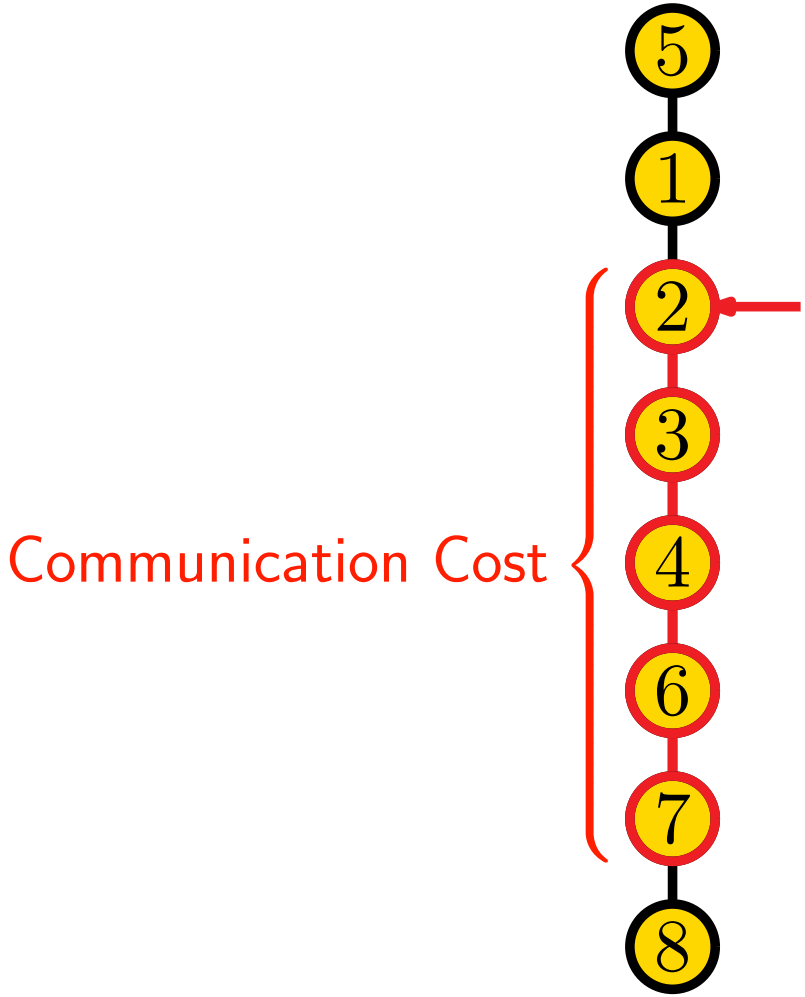
The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  $\rightarrow$ Dynamically Optimal!

The Network:

- Pairwise Access

Communicate (2,7)



A Simple[?] Network

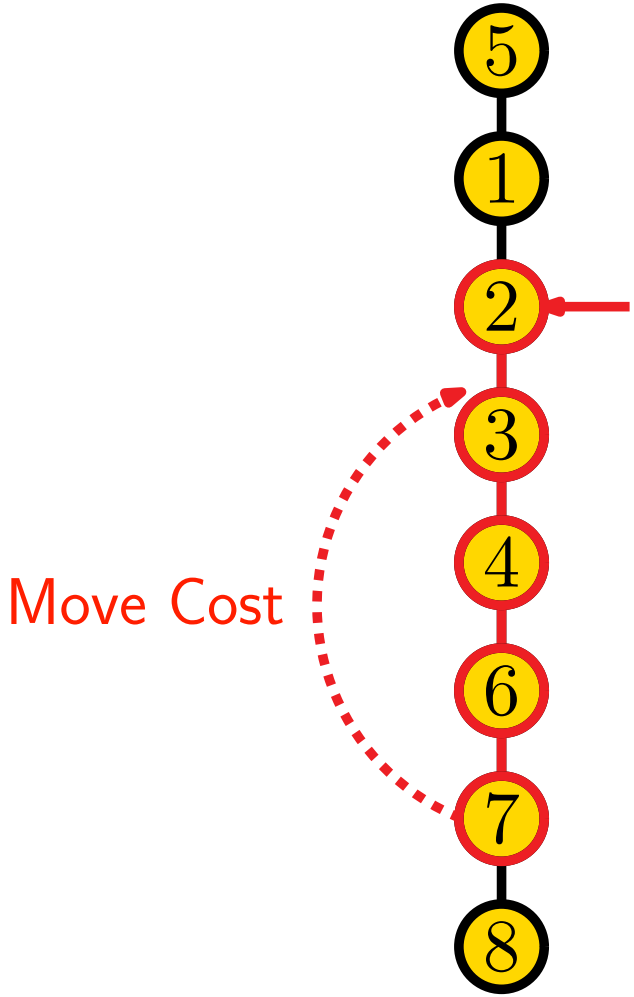
The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  → Dynamically Optimal!

The Network:

- Pairwise Access

Communicate (2,7)



A Simple[?] Network

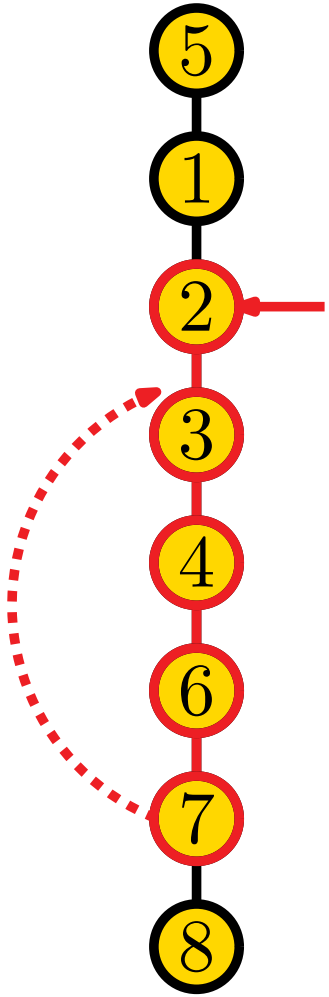
The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  → Dynamically Optimal!

The Network:

- Pairwise Access
- Move-to-Where?

Communicate (2,7)



A Simple[?] Network

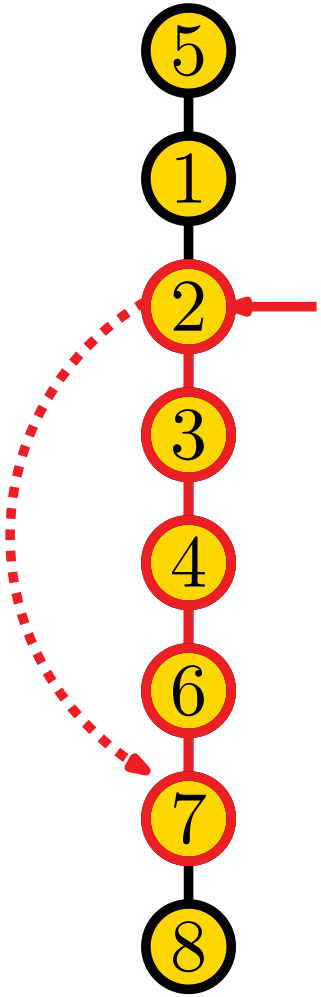
The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  → Dynamically Optimal!

The Network:

- Pairwise Access
- Move-to-Where?

Communicate (2,7)



A Simple[?] Network

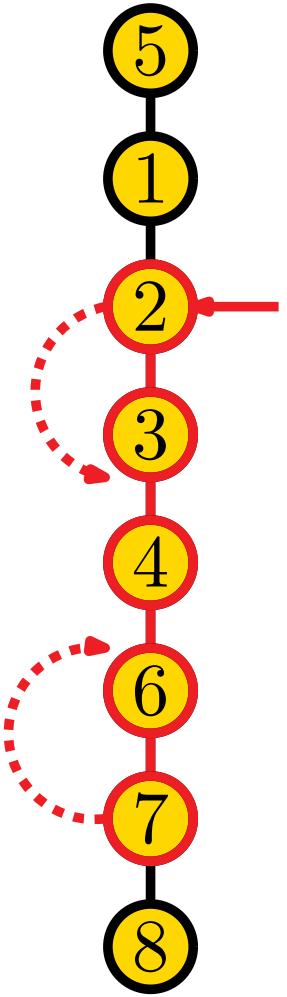
The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  → Dynamically Optimal!

The Network:

- Pairwise Access
- Move-to-Where?

Communicate (2,7)



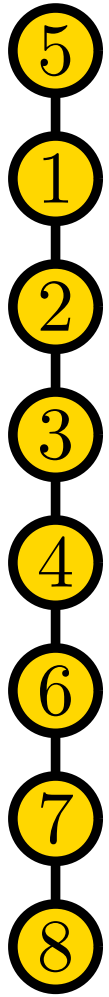
A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  $\rightarrow$ Dynamically Optimal!

The Network:

- Pairwise Access
- Move-to-Where?



A Simple[?] Network

The Dictionary:

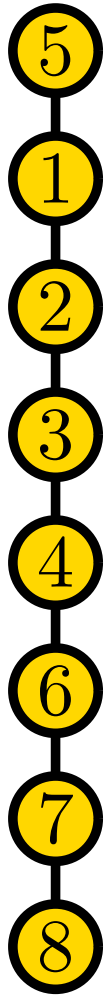
- Linked List
- Access in Front
- Move-to-Front Algorithm
  → Dynamically Optimal!

The Network:

- Pairwise Access
- Move-to-Where?

Not So Simple:

- Co-locating can Backfire



A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  → Dynamically Optimal!

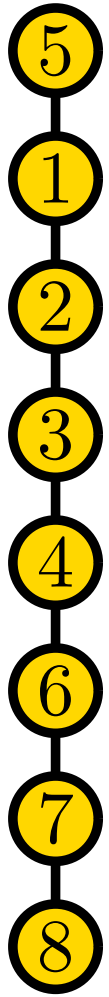
The Network:

- Pairwise Access
- Move-to-Where?

Not So Simple:

- Co-locating can Backfire

Communication:



A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  → Dynamically Optimal!

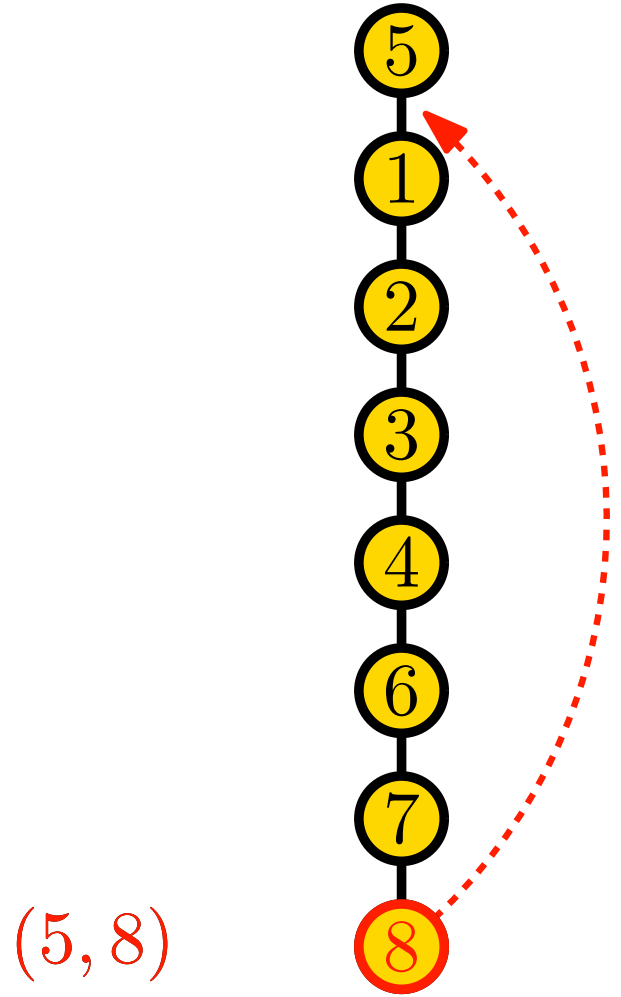
The Network:

- Pairwise Access
- Move-to-Where?

Not So Simple:

- Co-locating can Backfire

Communication:



A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  $\rightarrow$ Dynamically Optimal!

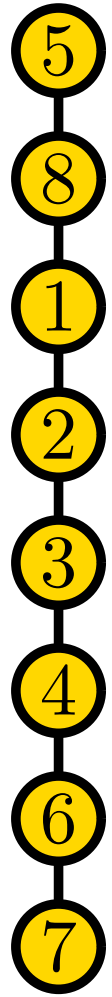
The Network:

- Pairwise Access
- Move-to-Where?

Not So Simple:

- Co-locating can Backfire

Communication:



$(5, 8)$

A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  → Dynamically Optimal!

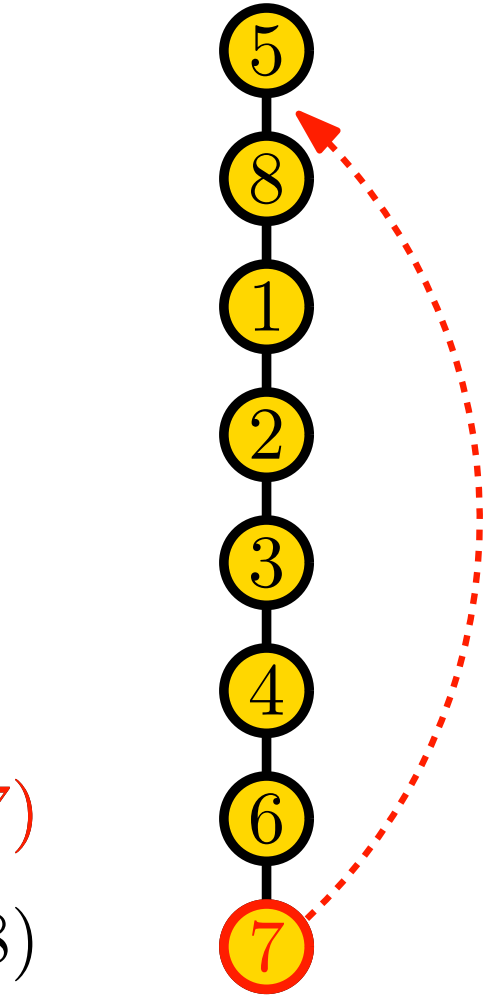
The Network:

- Pairwise Access
- Move-to-Where?

Not So Simple:

- Co-locating can Backfire

Communication:



$(5, 7)$

$(5, 8)$

A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  $\rightarrow$ Dynamically Optimal!

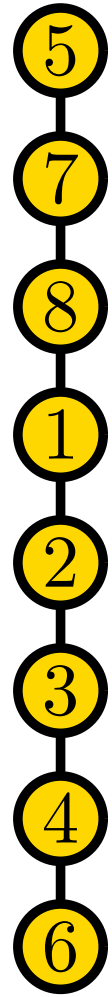
The Network:

- Pairwise Access
- Move-to-Where?

Not So Simple:

- Co-locating can Backfire

Communication:



$(5, 7)$

$(5, 8)$

A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  $\rightarrow$ Dynamically Optimal!

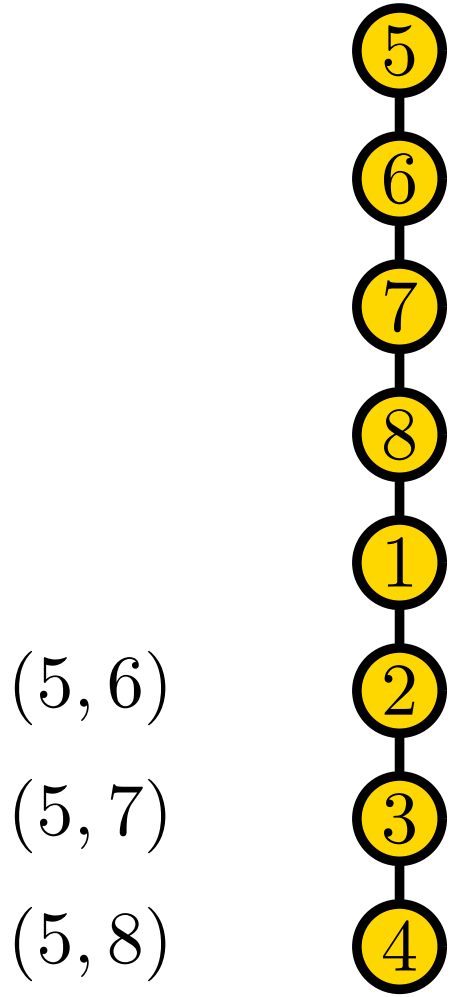
The Network:

- Pairwise Access
- Move-to-Where?

Not So Simple:

- Co-locating can Backfire

Communication:



A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  $\rightarrow$ Dynamically Optimal!

The Network:

- Pairwise Access
- Move-to-Where?

Not So Simple:

- Co-locating can Backfire

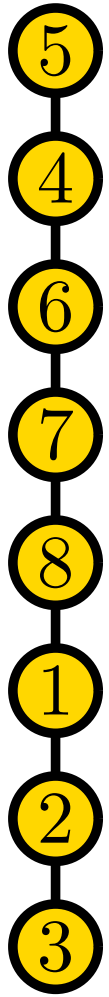
Communication:

$(5, 4)$

$(5, 6)$

$(5, 7)$

$(5, 8)$



A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  $\rightarrow$ Dynamically Optimal!

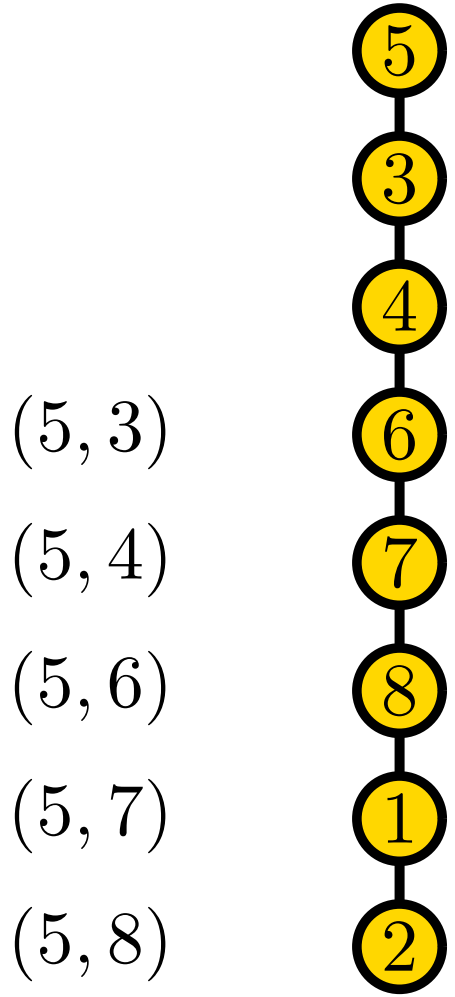
The Network:

- Pairwise Access
- Move-to-Where?

Not So Simple:

- Co-locating can Backfire

Communication:



A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  $\rightarrow$ Dynamically Optimal!

The Network:

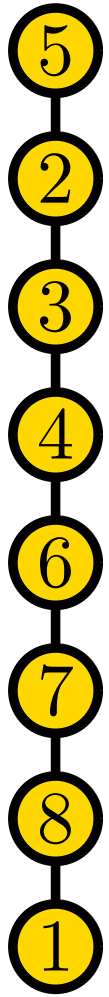
- Pairwise Access
- Move-to-Where?

Not So Simple:

- Co-locating can Backfire

Communication:

$(5, 2)$
$(5, 3)$
$(5, 4)$
$(5, 6)$
$(5, 7)$
$(5, 8)$



A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  $\rightarrow$ Dynamically Optimal!

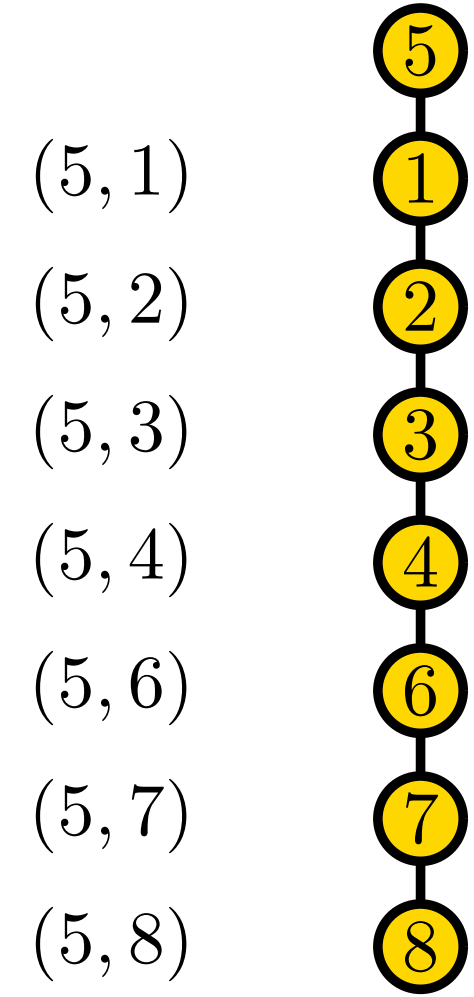
The Network:

- Pairwise Access
- Move-to-Where?

Not So Simple:

- Co-locating can Backfire

Communication:



A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  → Dynamically Optimal!

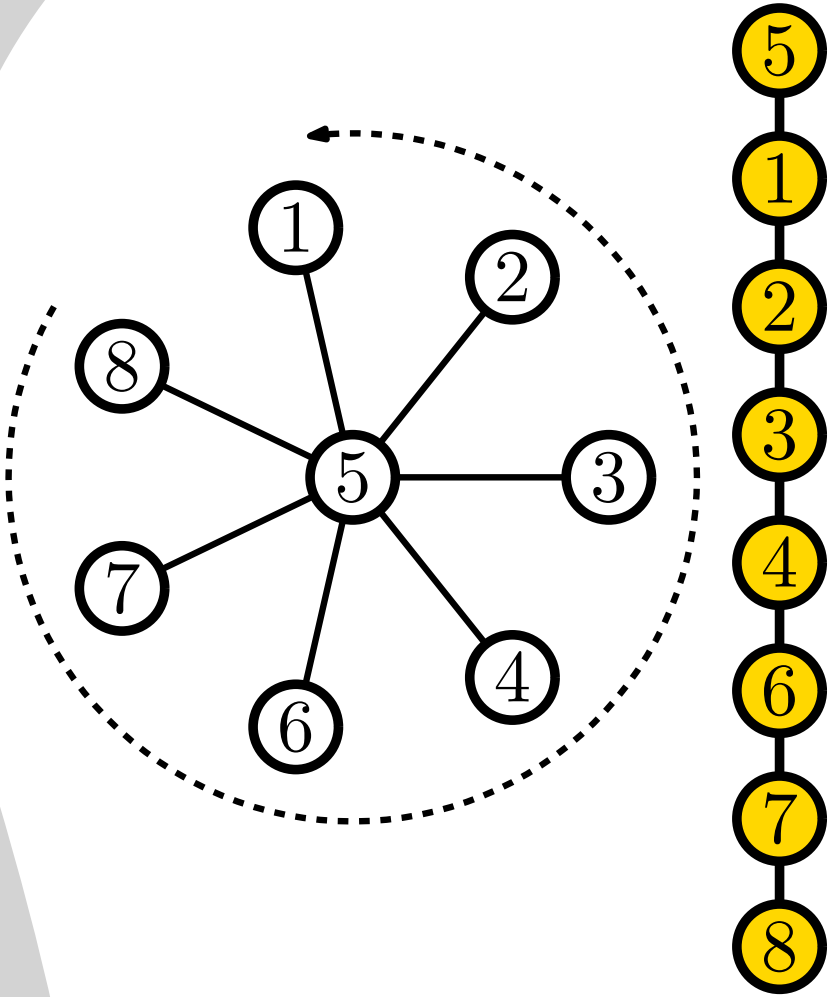
The Network:

- Pairwise Access
- Move-to-Where?

Not So Simple:

- Co-locating can Backfire
- Wheel can be Everywhere

Communication:



A Simple[?] Network

The Dictionary:

- Linked List
- Access in Front
- Move-to-Front Algorithm
  → Dynamically Optimal!

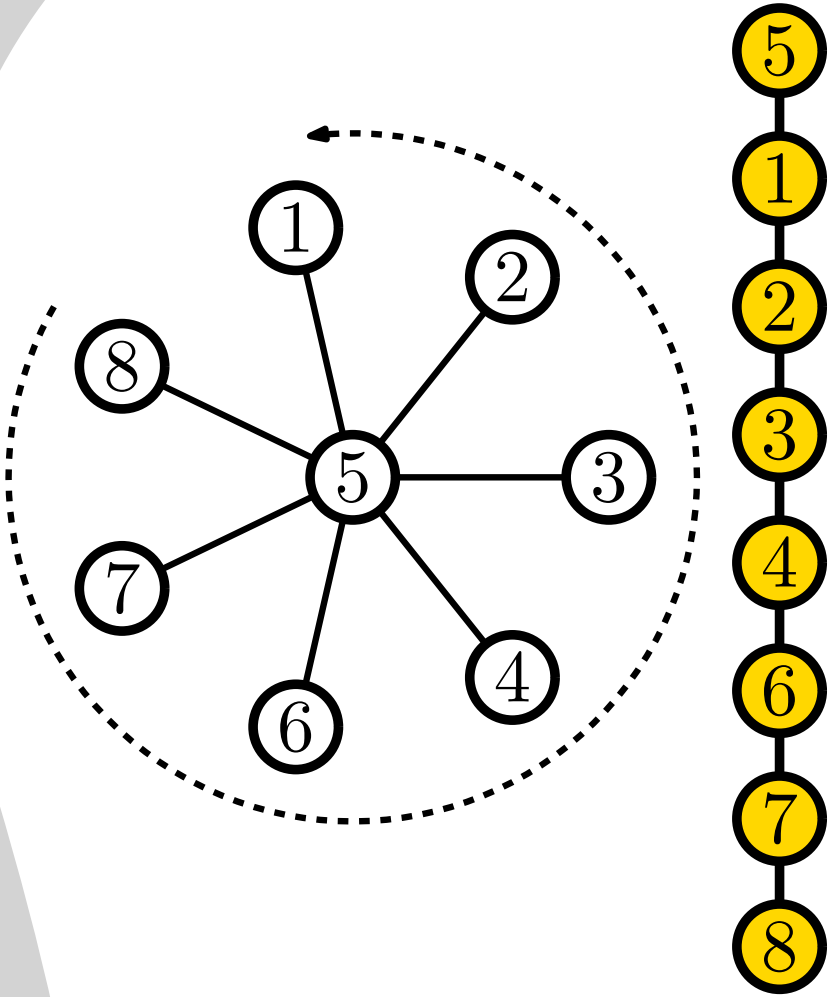
The Network:

- Pairwise Access
- Move-to-Where?

Not So Simple:

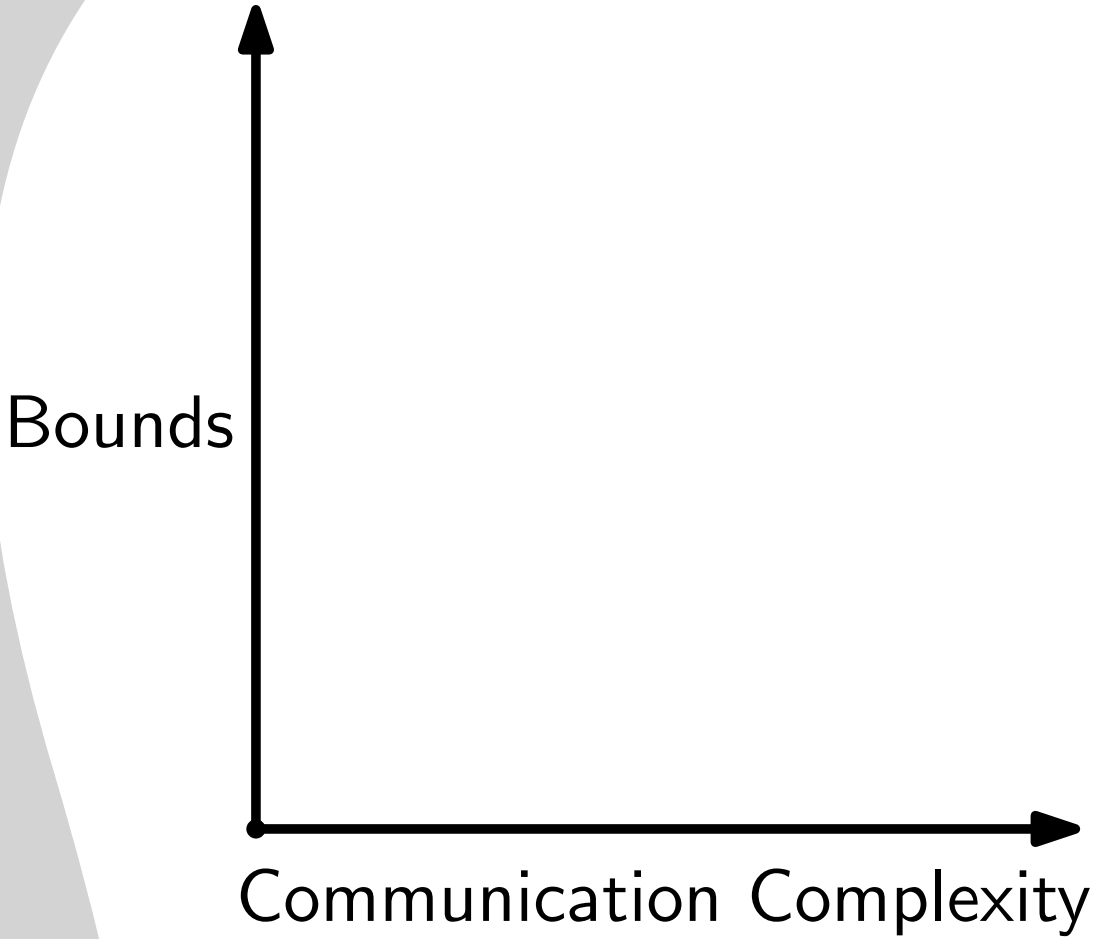
- Co-locating can Backfire
- Wheel can be Everywhere
- Same Conclusion by Olver et al.

Communication:



Results

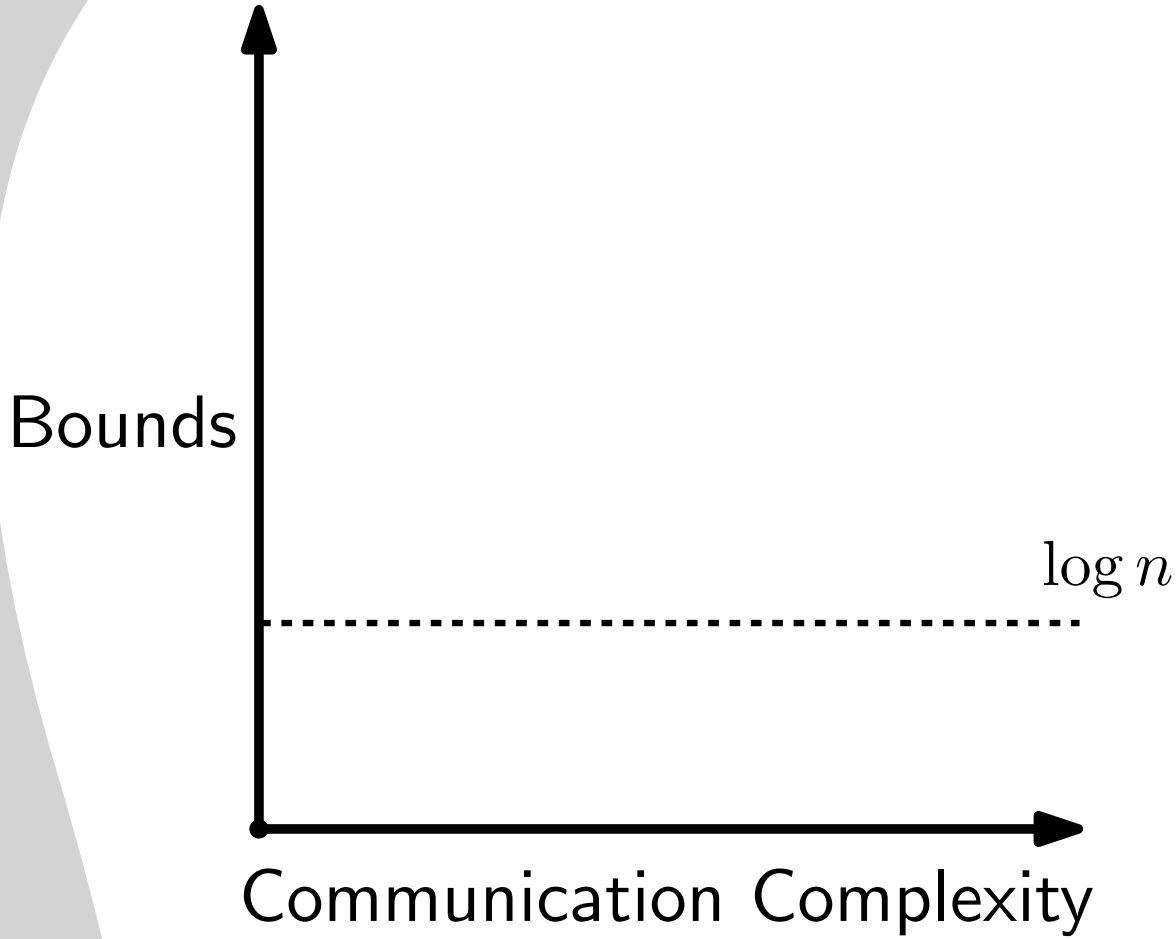
Results



Results

Grid Networks:

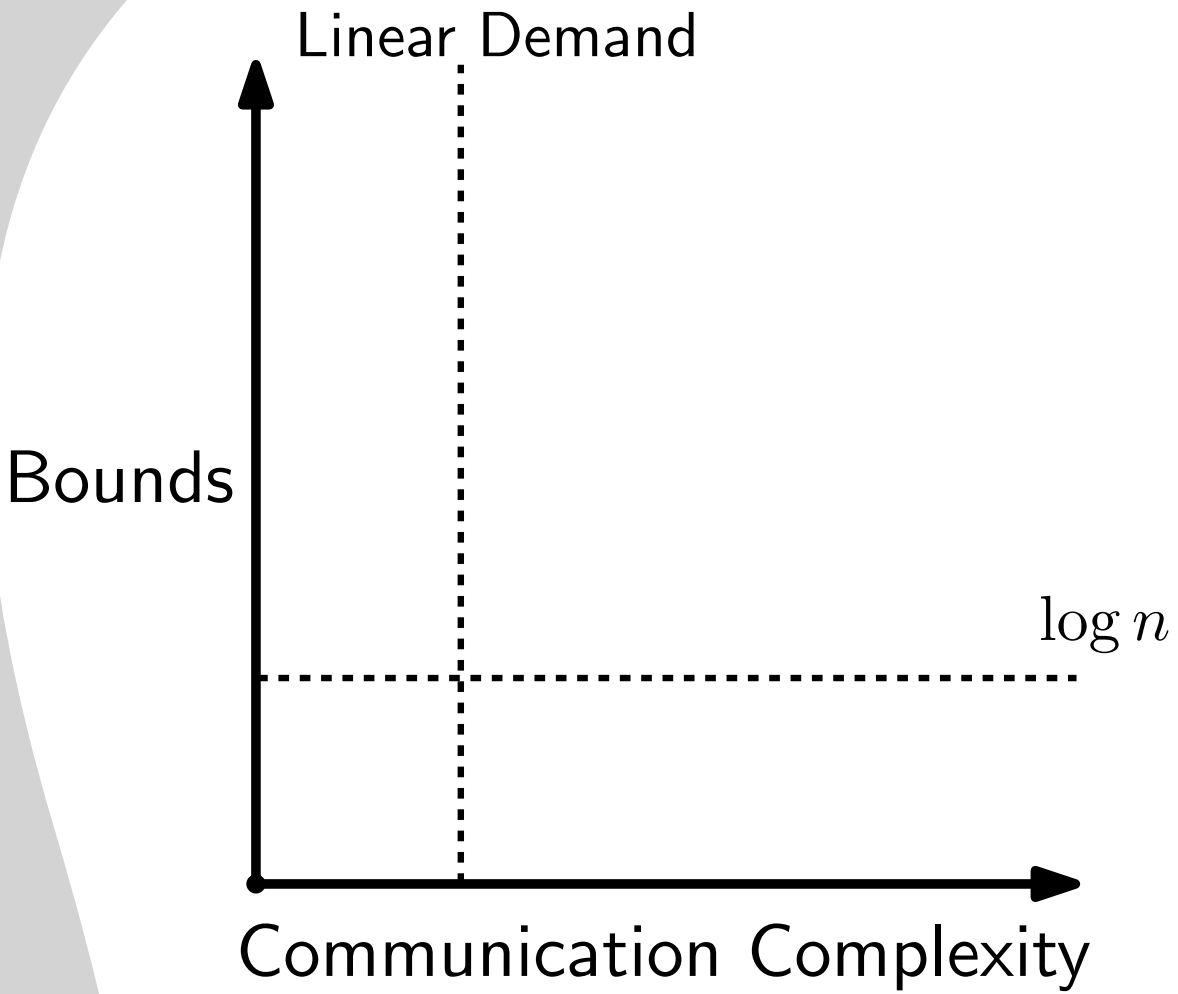
- Bounds on Competitive Ratio



Results

Grid Networks:

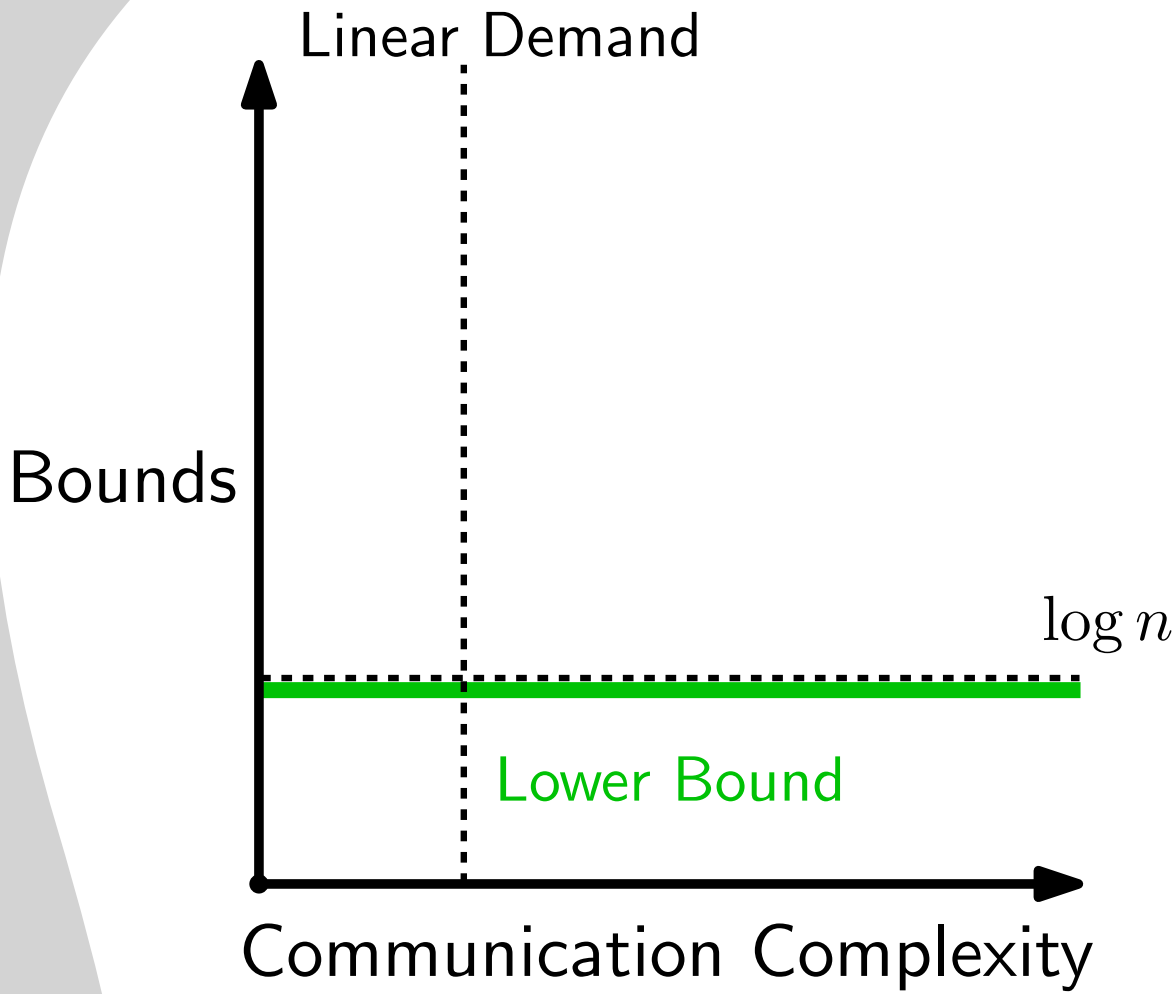
- Bounds on Competitive Ratio
- Restricted Communication



Results

Grid Networks:

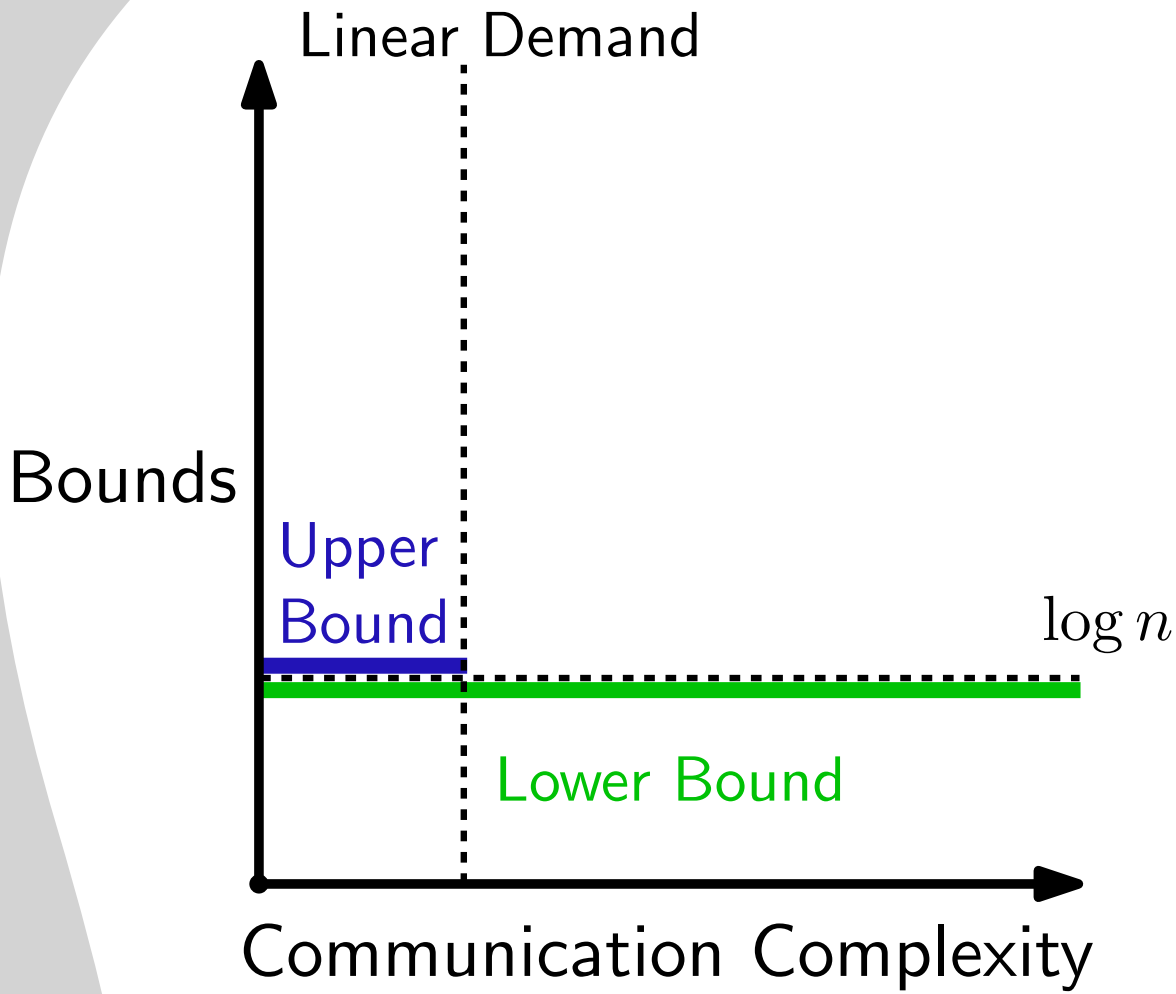
- Bounds on Competitive Ratio
- Restricted Communication



Results

Grid Networks:

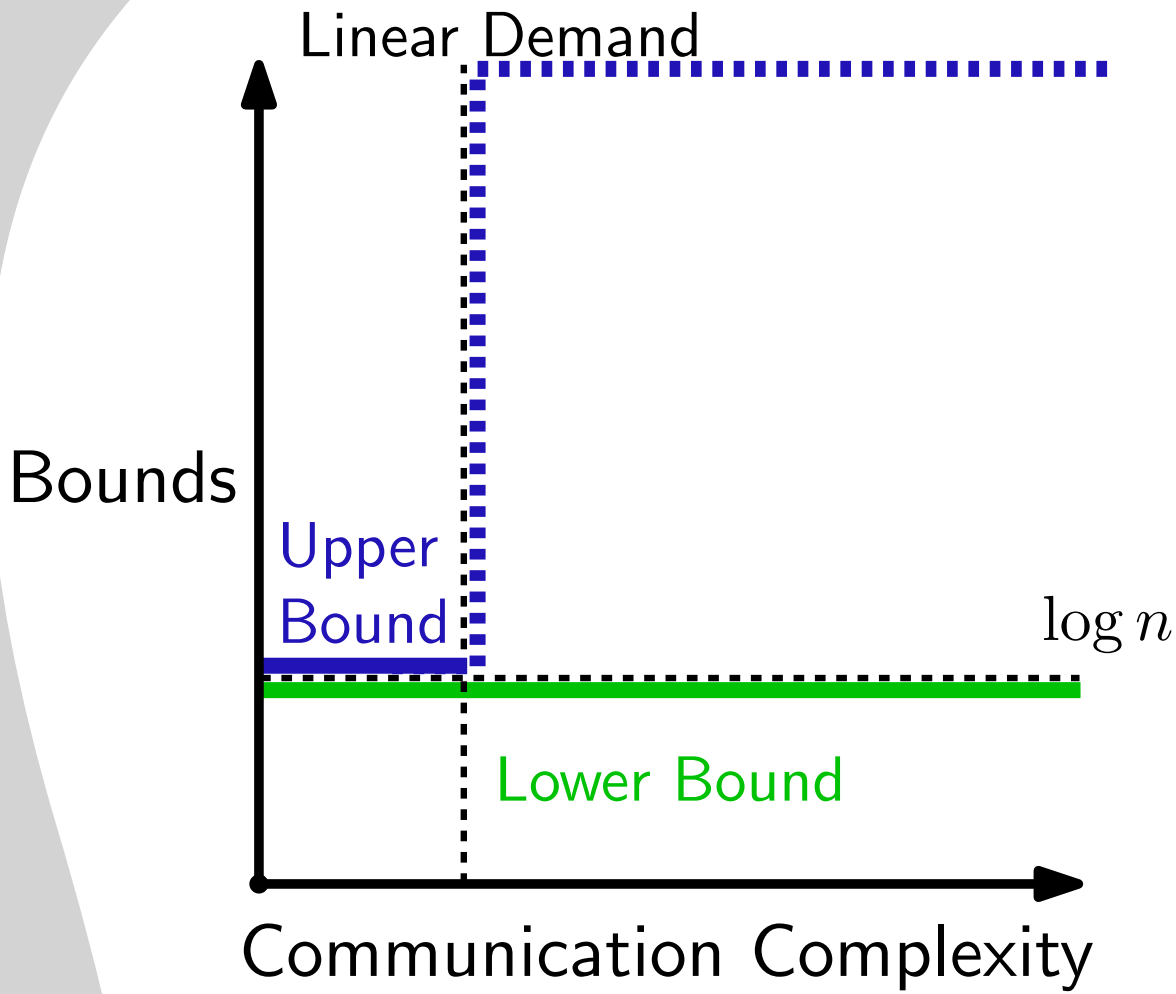
- Bounds on Competitive Ratio
- Restricted Communication



Results

Grid Networks:

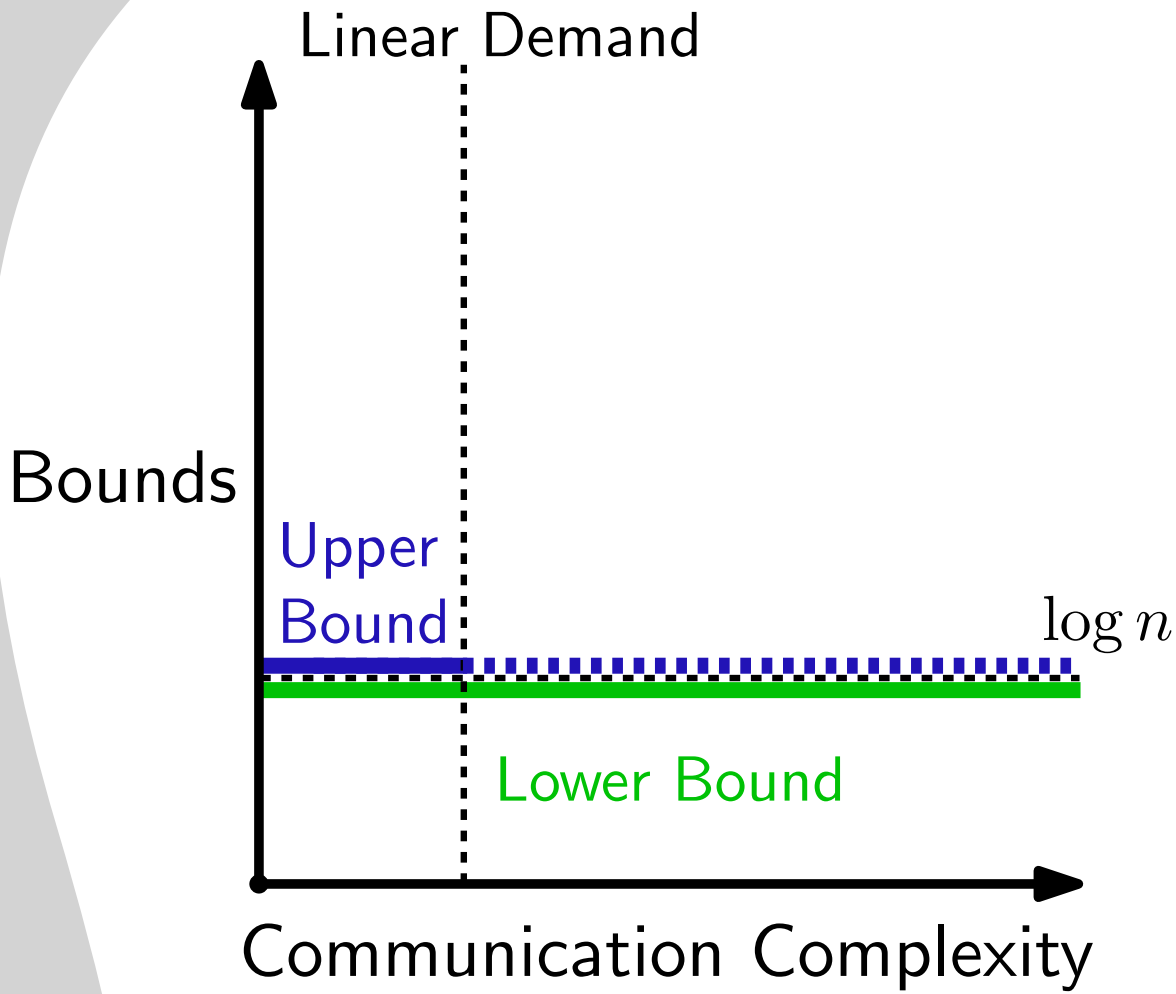
- Bounds on Competitive Ratio
- Restricted Communication



Results

Grid Networks:

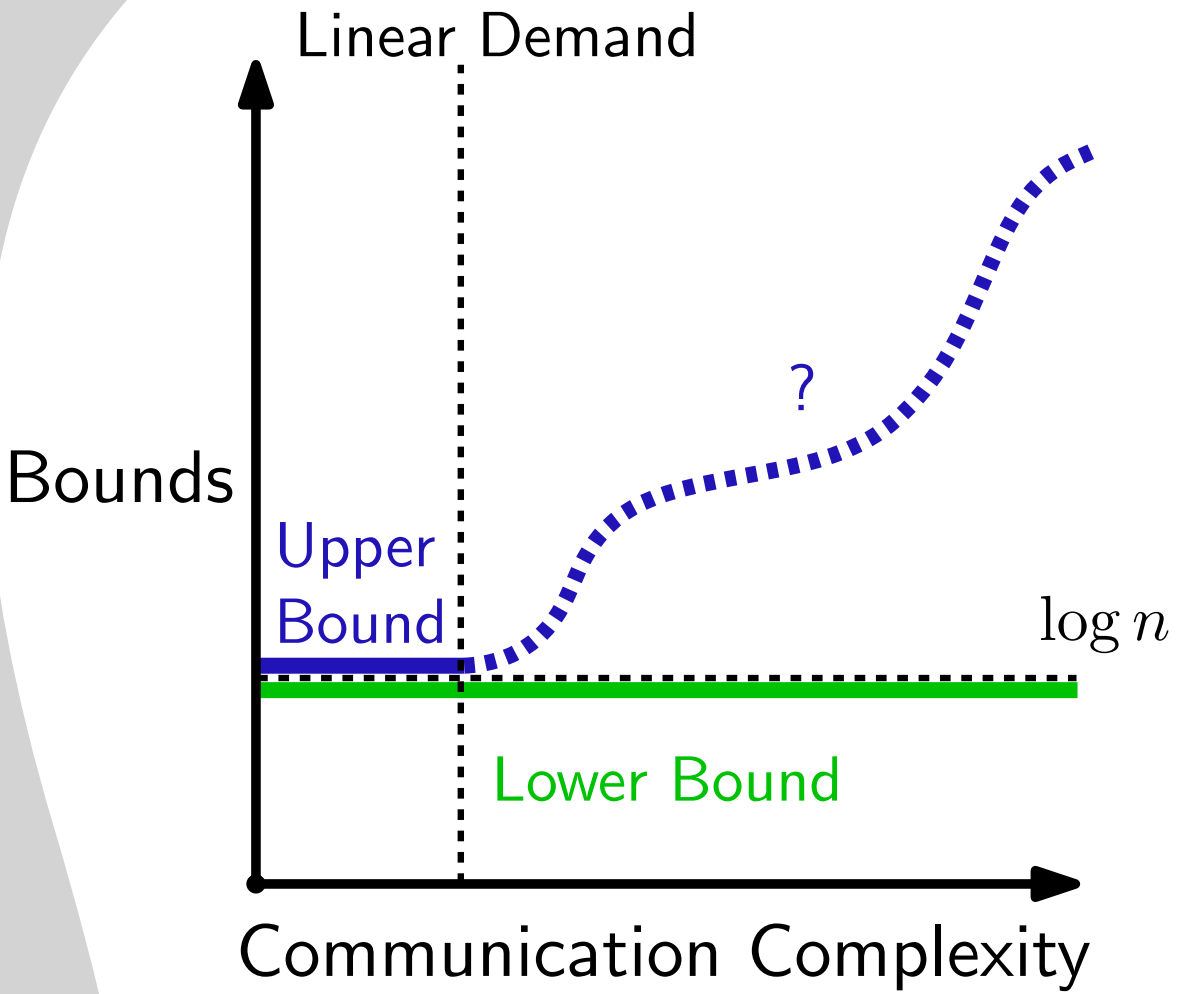
- Bounds on Competitive Ratio
- Restricted Communication



Results

Grid Networks:

- Bounds on Competitive Ratio
- Restricted Communication



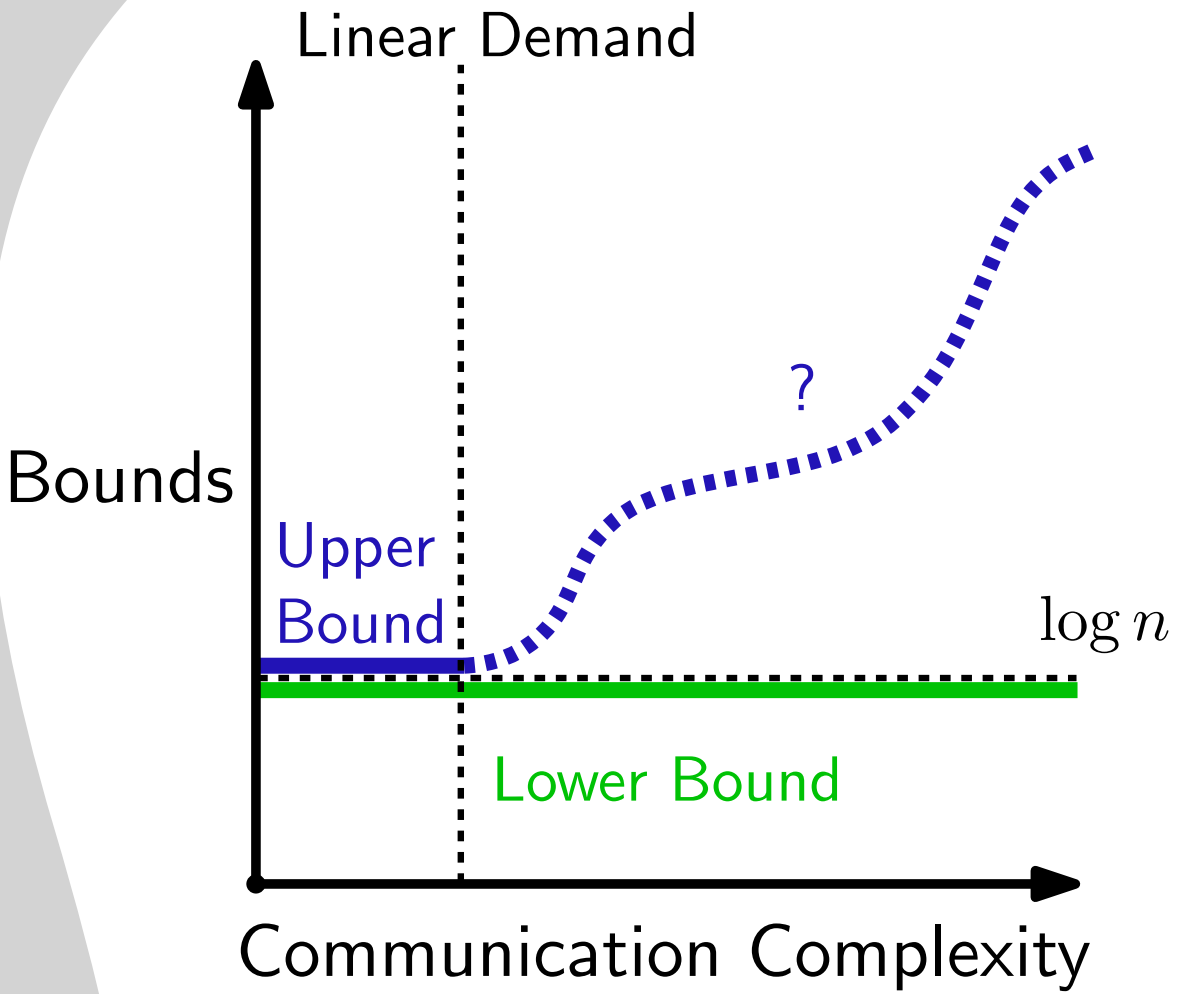
Results

Grid Networks:

- Bounds on Competitive Ratio
- Restricted Communication

Line Networks:

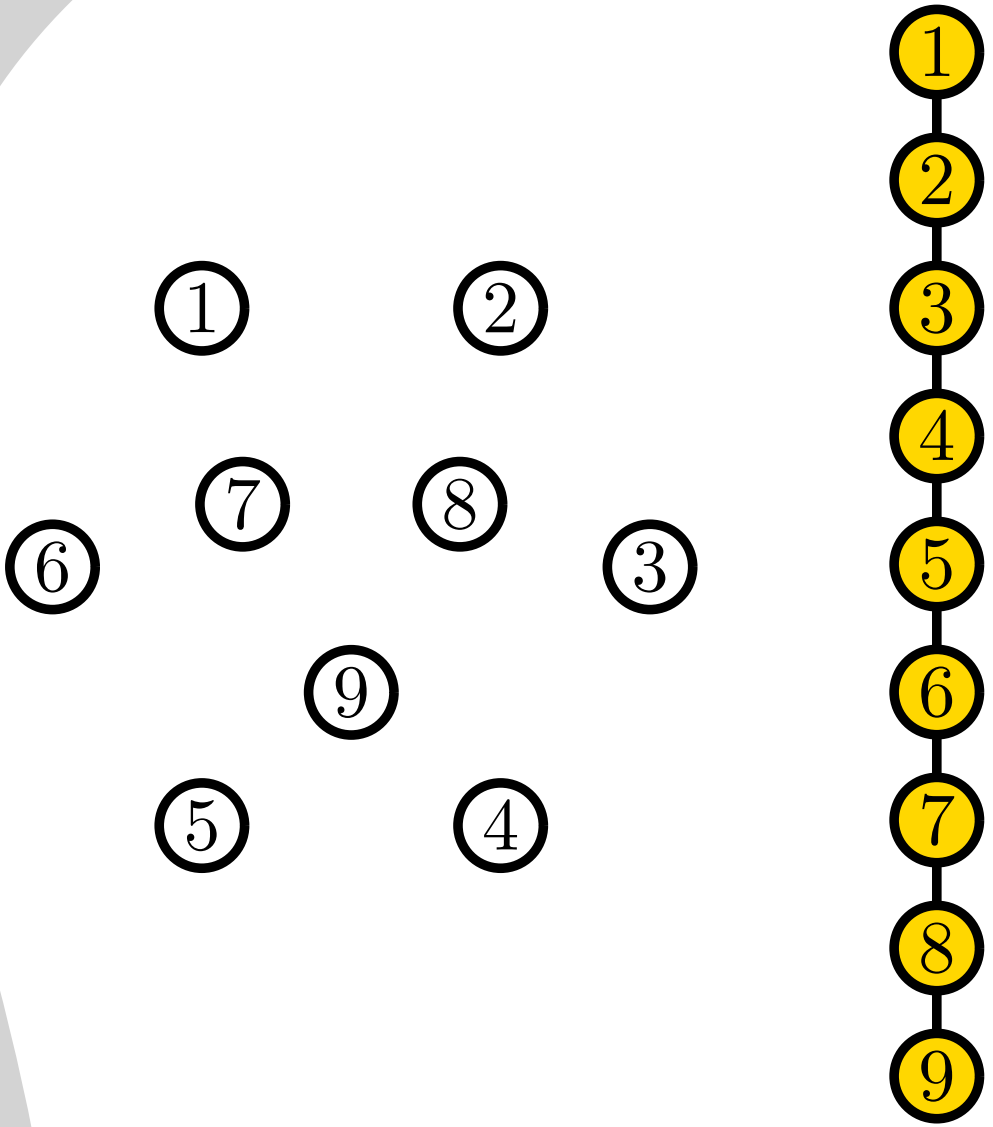
- Distributed Implementation



Lower Bound

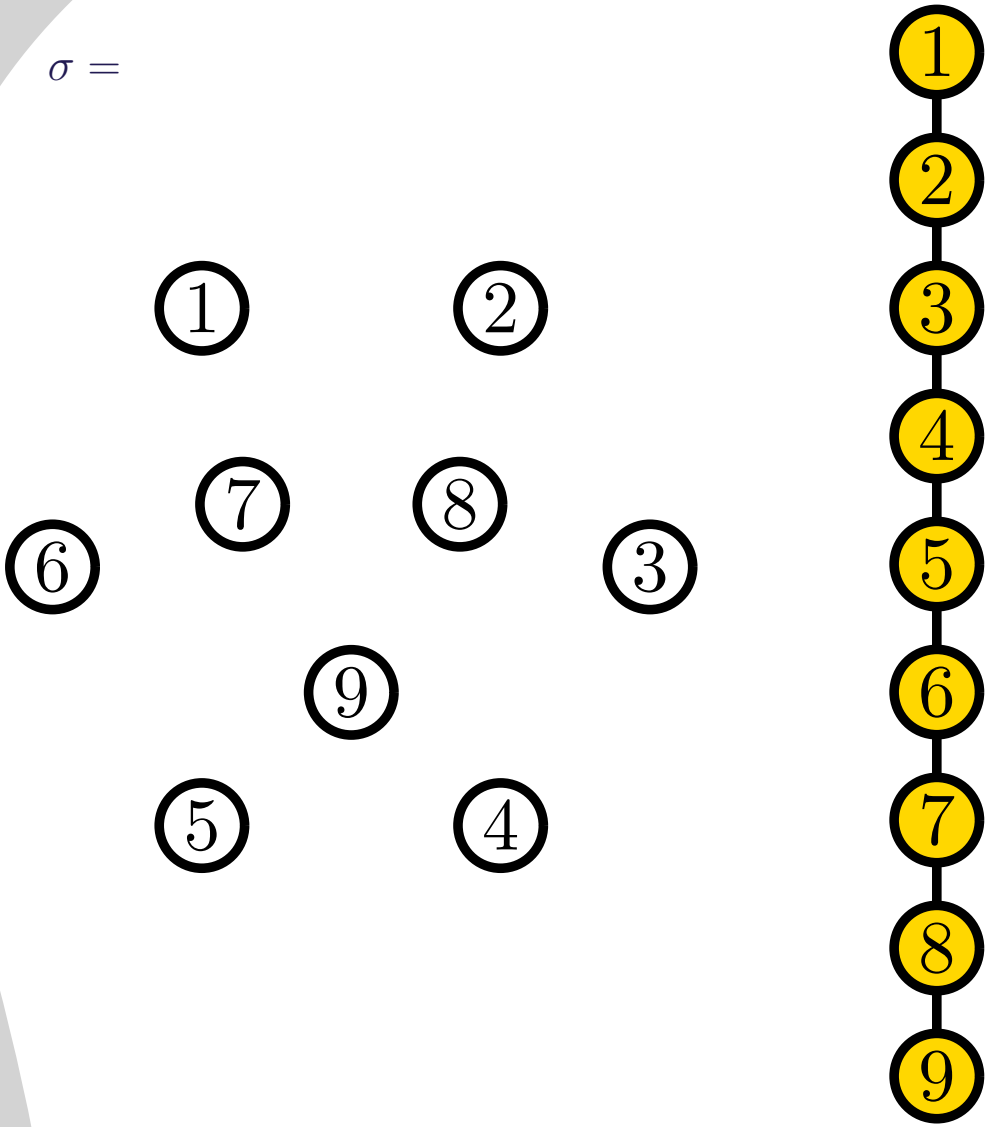
Lower Bound

Lower Bound



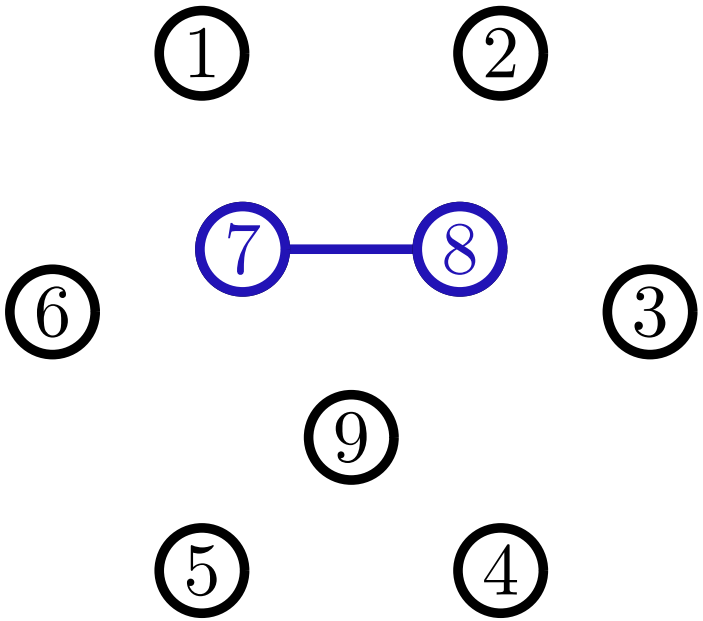
Lower Bound

$\sigma =$



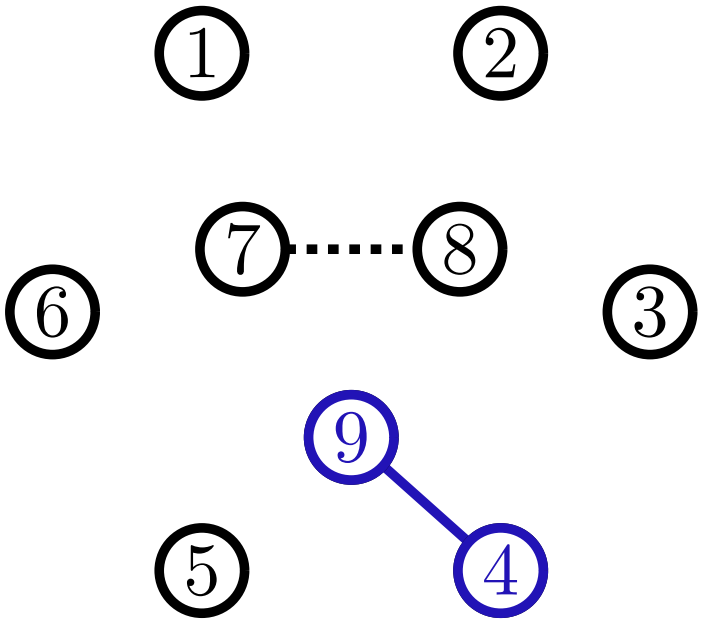
Lower Bound

$\sigma = (7, 8)$



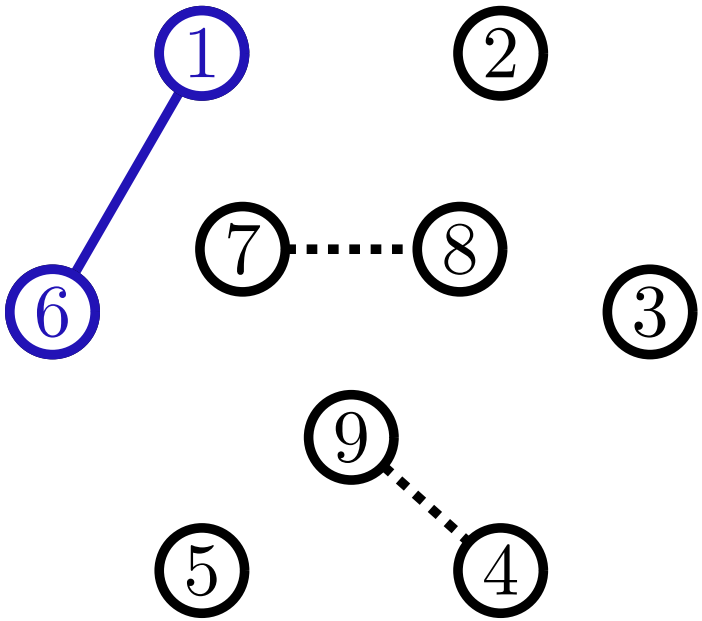
Lower Bound

$\sigma = (7,8)(4,9)$



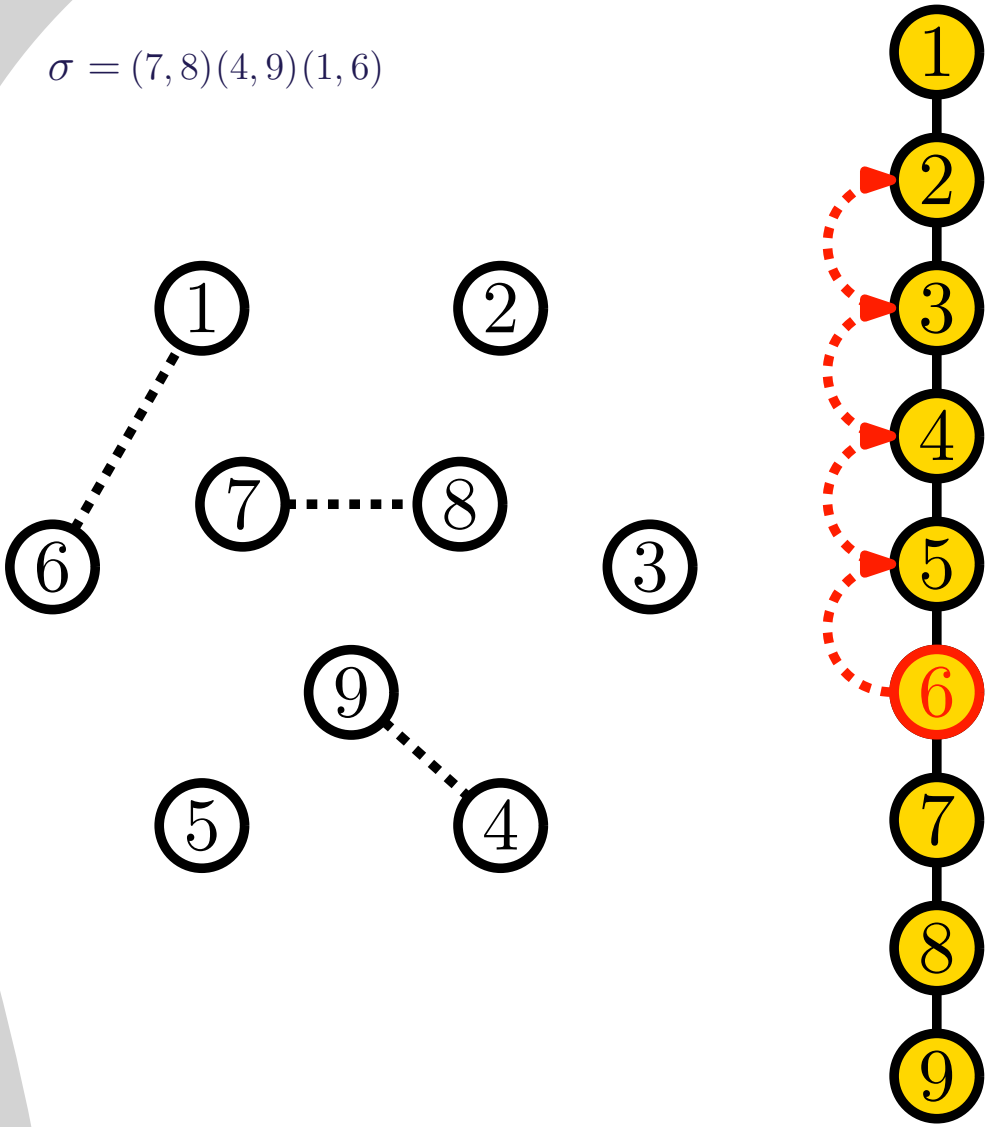
Lower Bound

$\sigma = (7,8)(4,9)(1,6)$



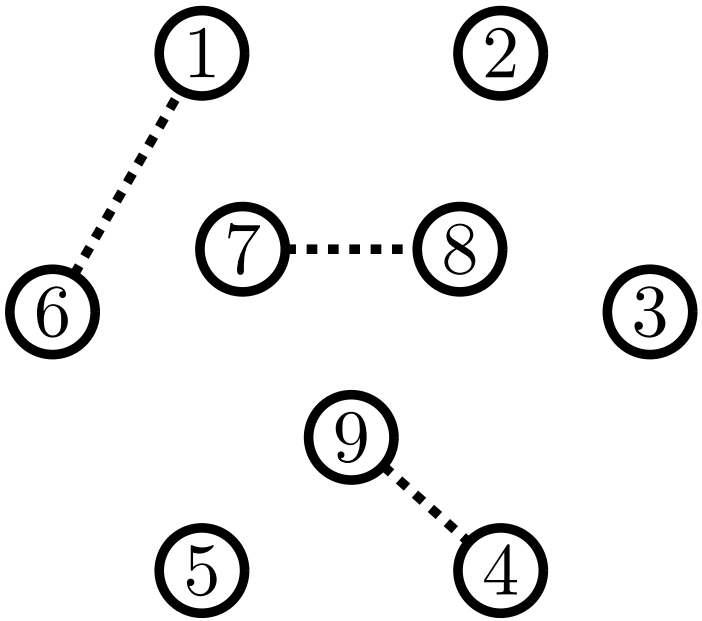
Lower Bound

$\sigma = (7,8)(4,9)(1,6)$



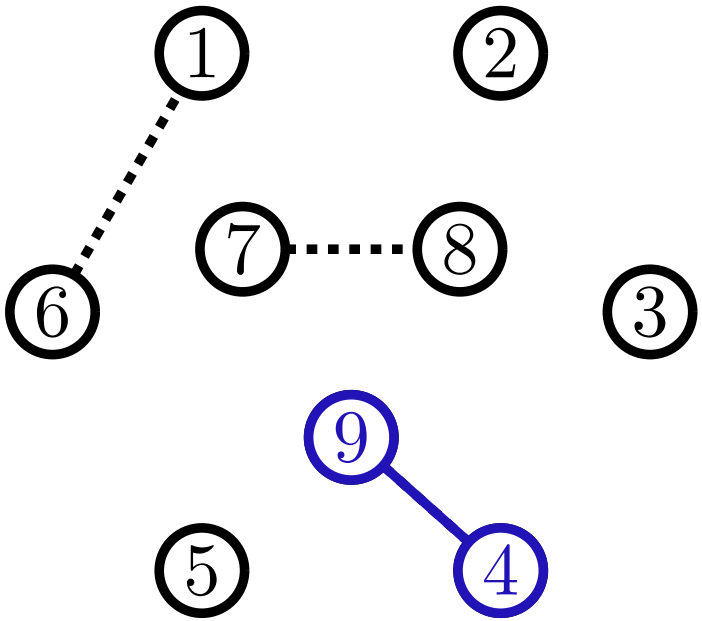
Lower Bound

$\sigma = (7,8)(4,9)(1,6)$



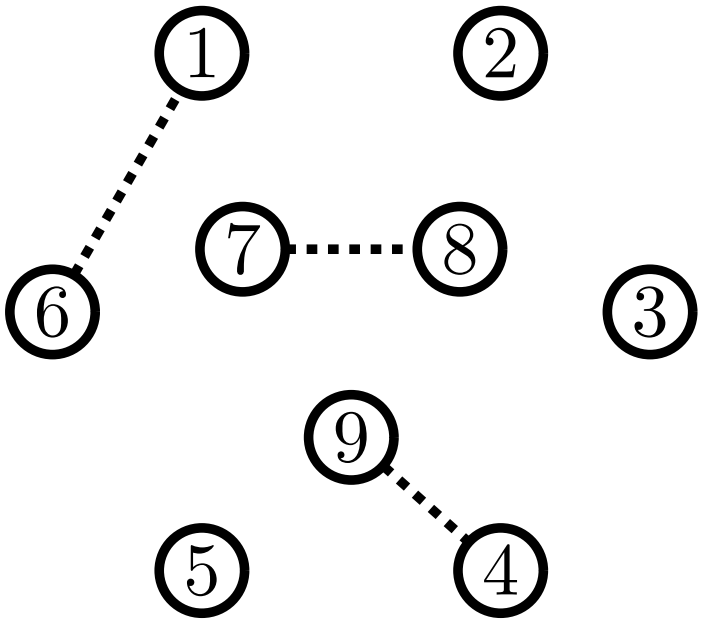
Lower Bound

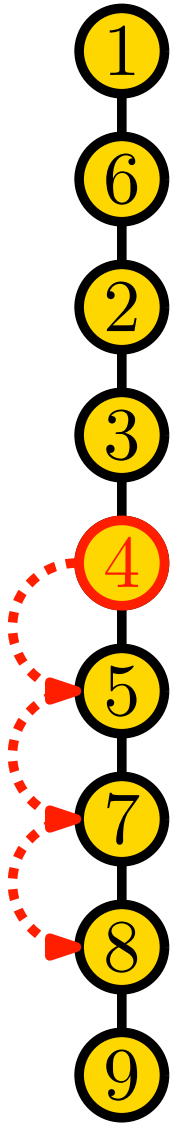
$\sigma = (7,8)(4,9)(1,6)(4,9)$



Lower Bound

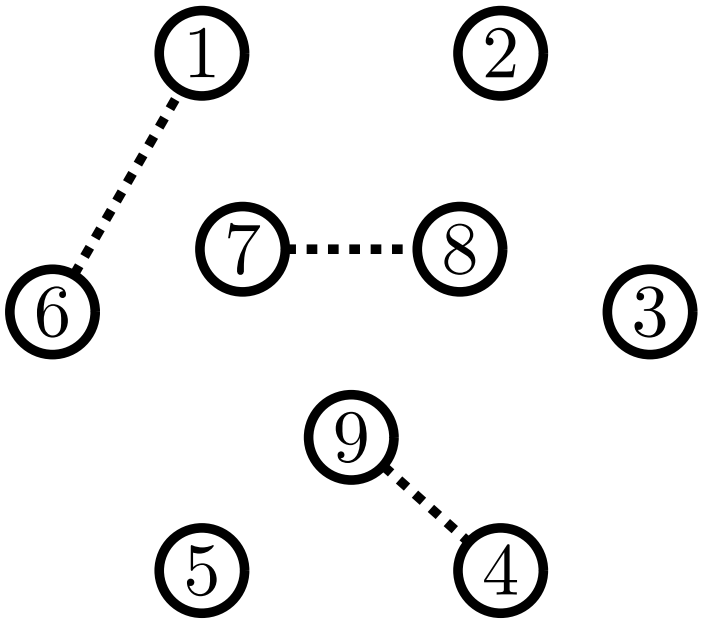
$\sigma = (7,8)(4,9)(1,6)(4,9)$





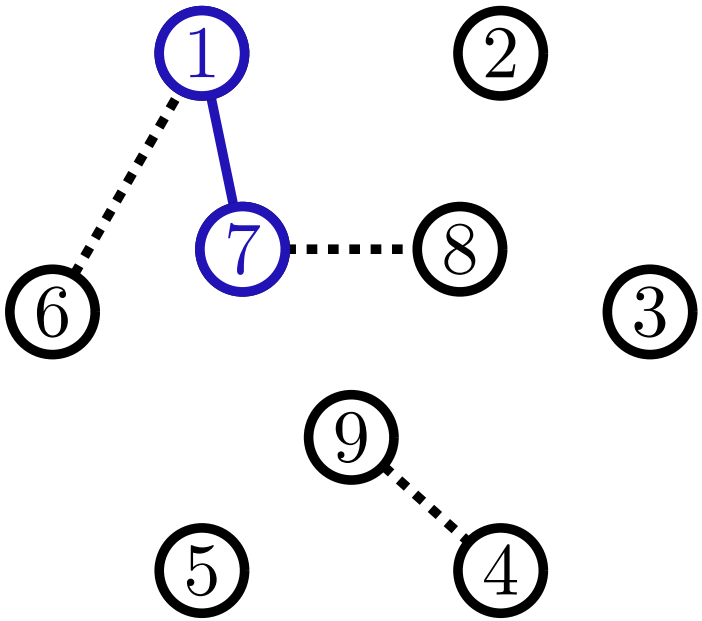
Lower Bound

$\sigma = (7,8)(4,9)(1,6)(4,9)$



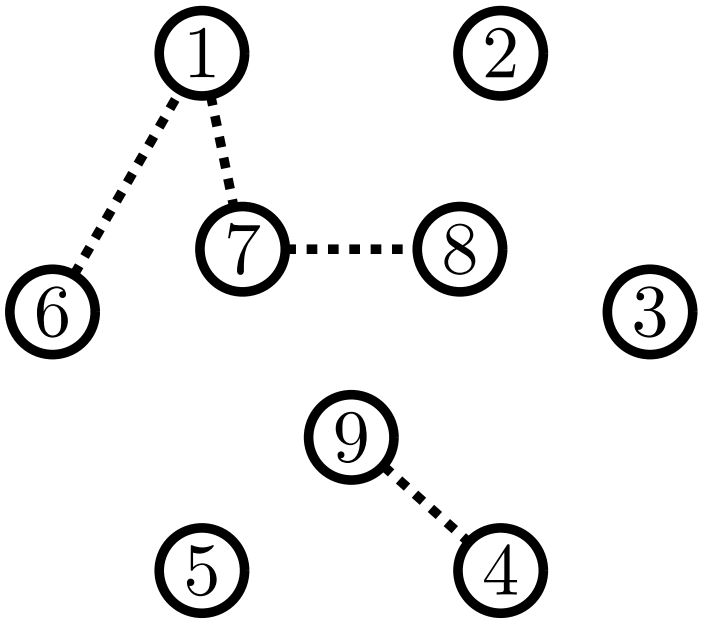
Lower Bound

$\sigma = (7,8)(4,9)(1,6)(4,9)(1,7)$



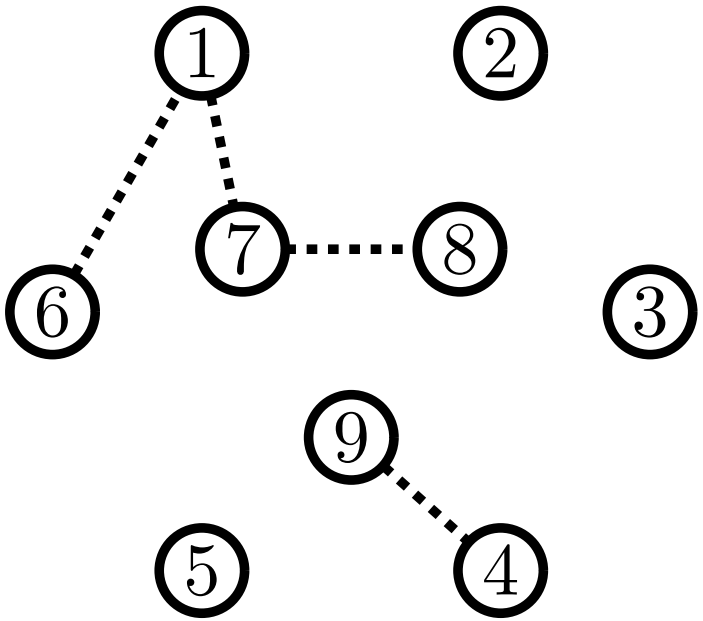
Lower Bound

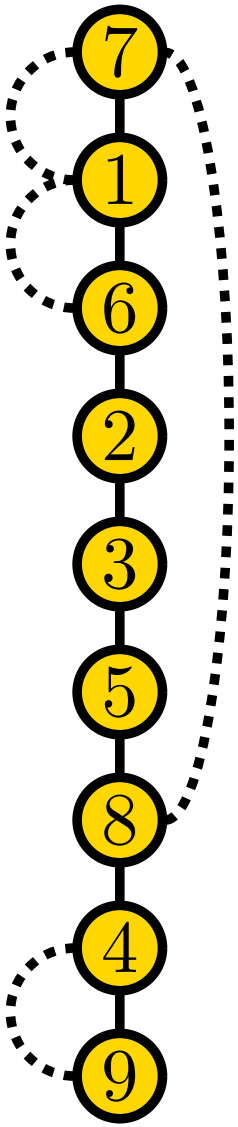
$\sigma = (7,8)(4,9)(1,6)(4,9)(1,7)$



Lower Bound

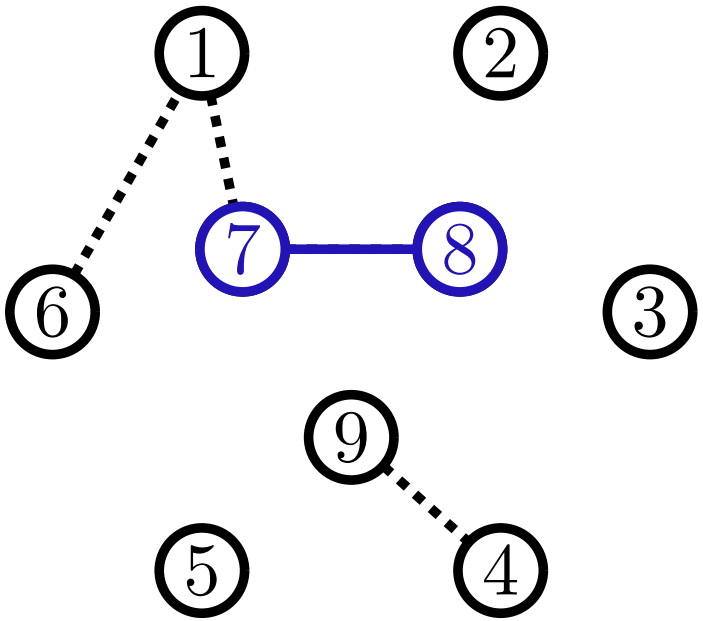
$\sigma = (7,8)(4,9)(1,6)(4,9)(1,7)$





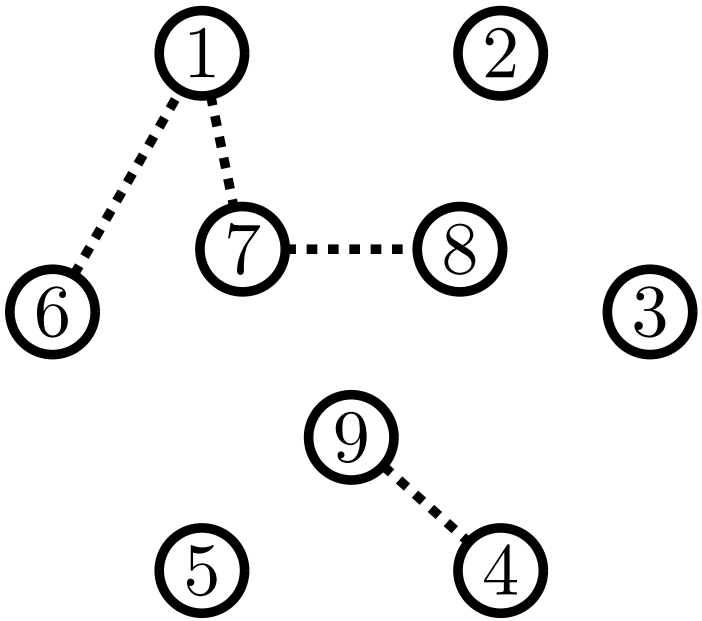
Lower Bound

$\sigma = (7,8)(4,9)(1,6)(4,9)(1,7)(7,8)$



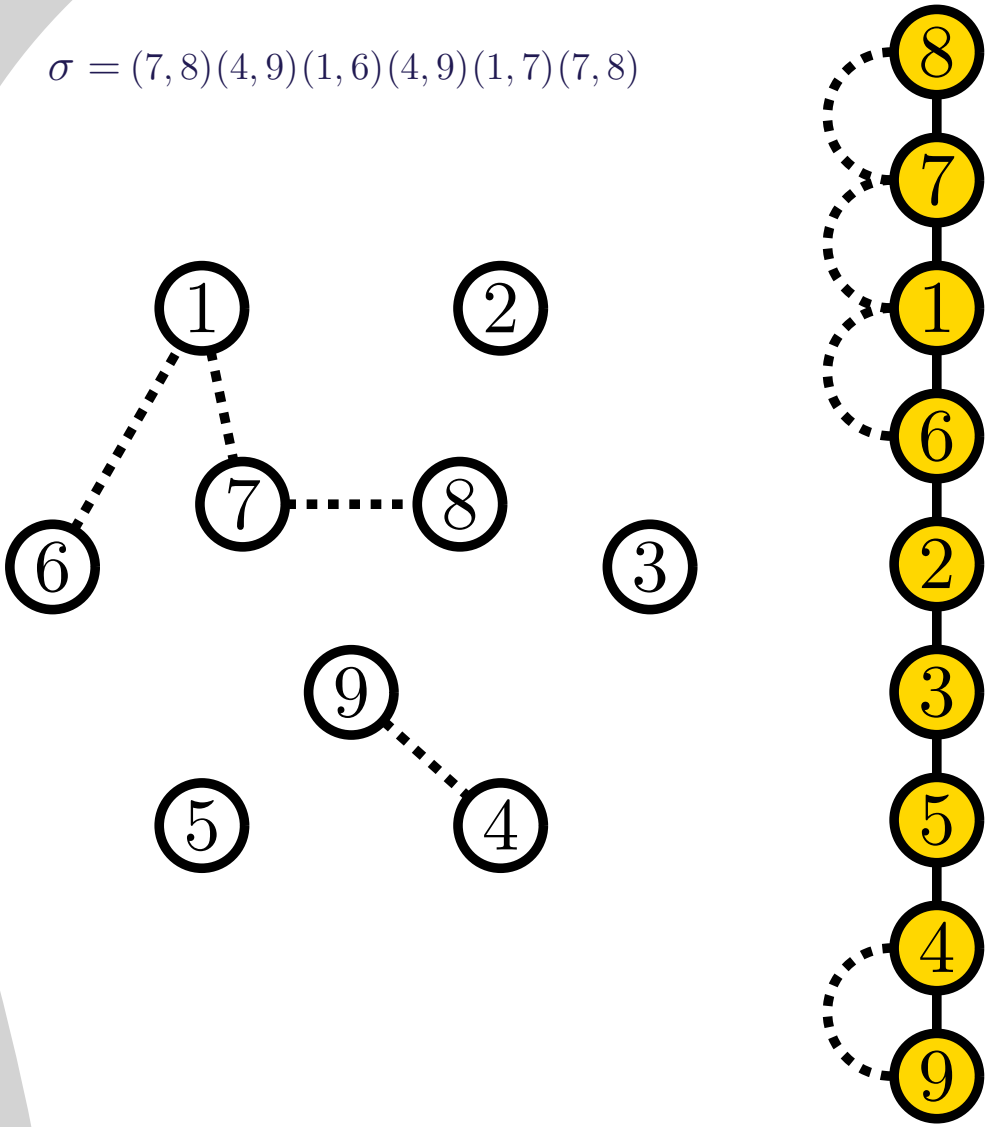
Lower Bound

$\sigma = (7,8)(4,9)(1,6)(4,9)(1,7)(7,8)$



Lower Bound

$\sigma = (7,8)(4,9)(1,6)(4,9)(1,7)(7,8)$

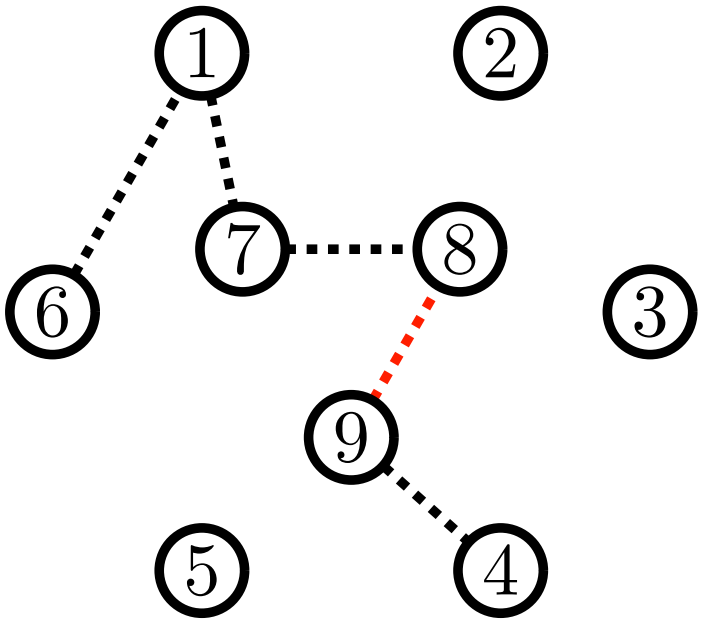


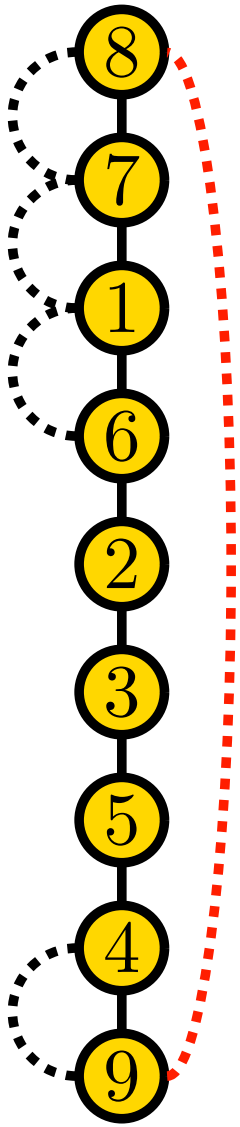
Lower Bound

The Strategy:

- Exploit Bad Edges

$\sigma = (7,8)(4,9)(1,6)(4,9)(1,7)(7,8)$

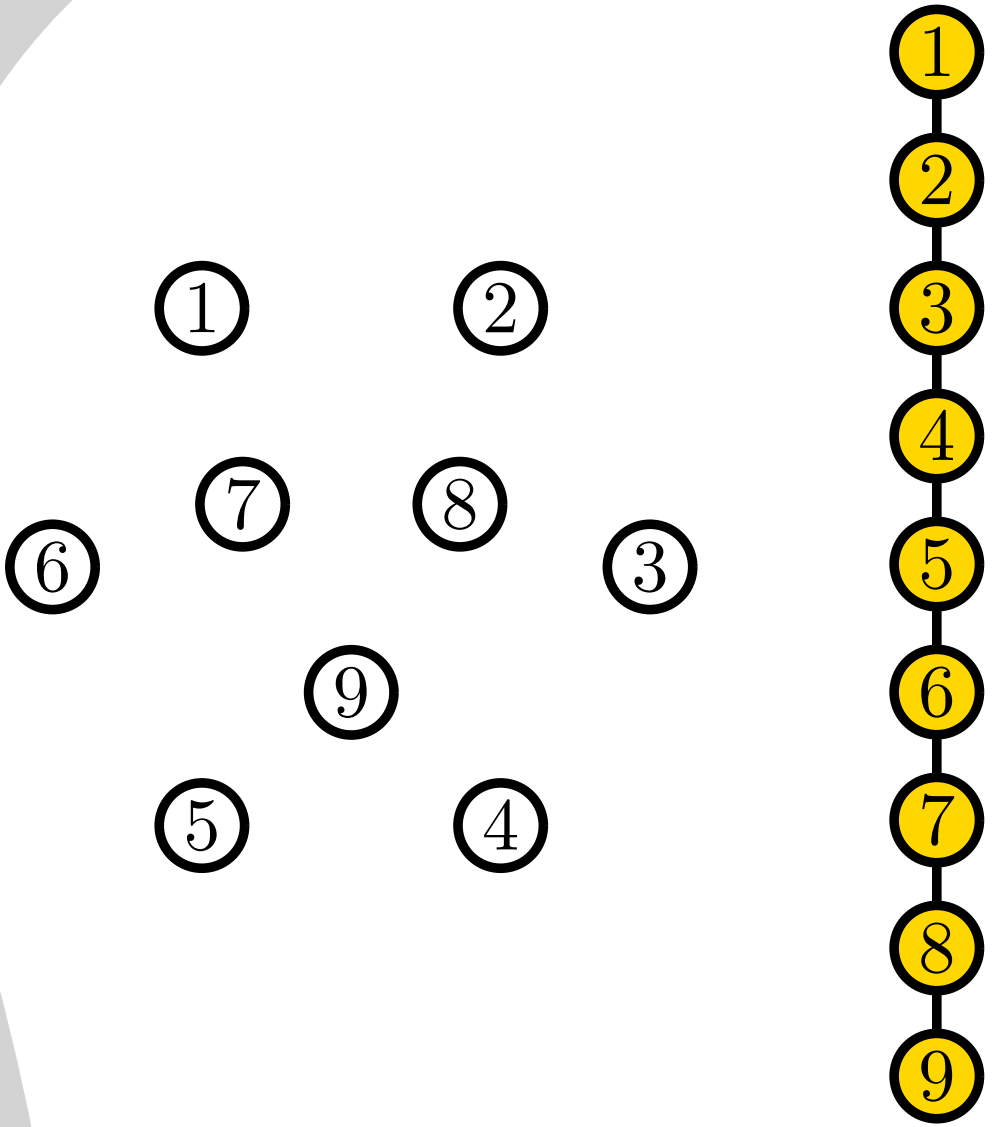




Lower Bound

The Strategy:

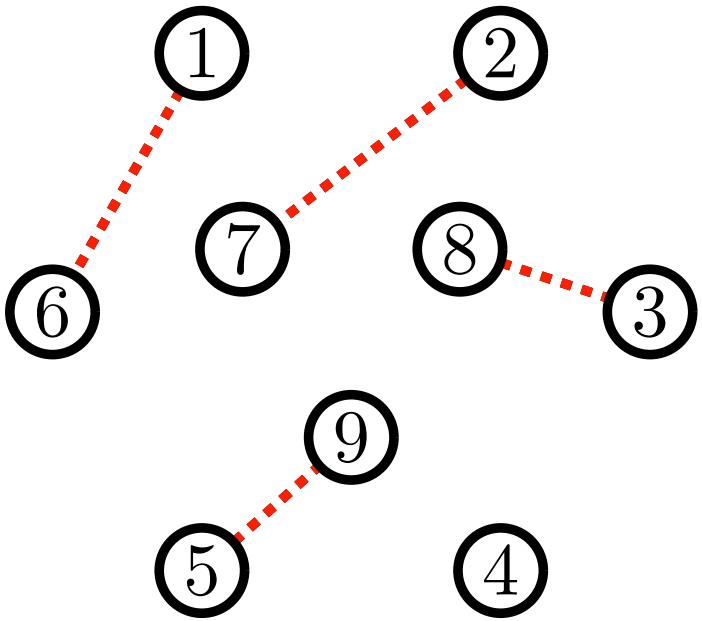
- Exploit Bad Edges

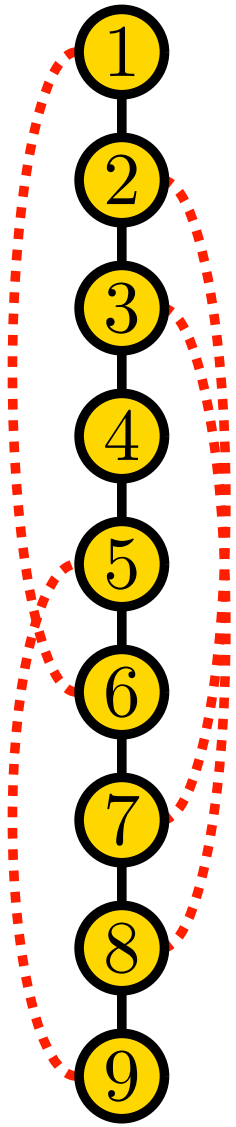


Lower Bound

The Strategy:

- Exploit Bad Edges
- Introduce Bad Matching

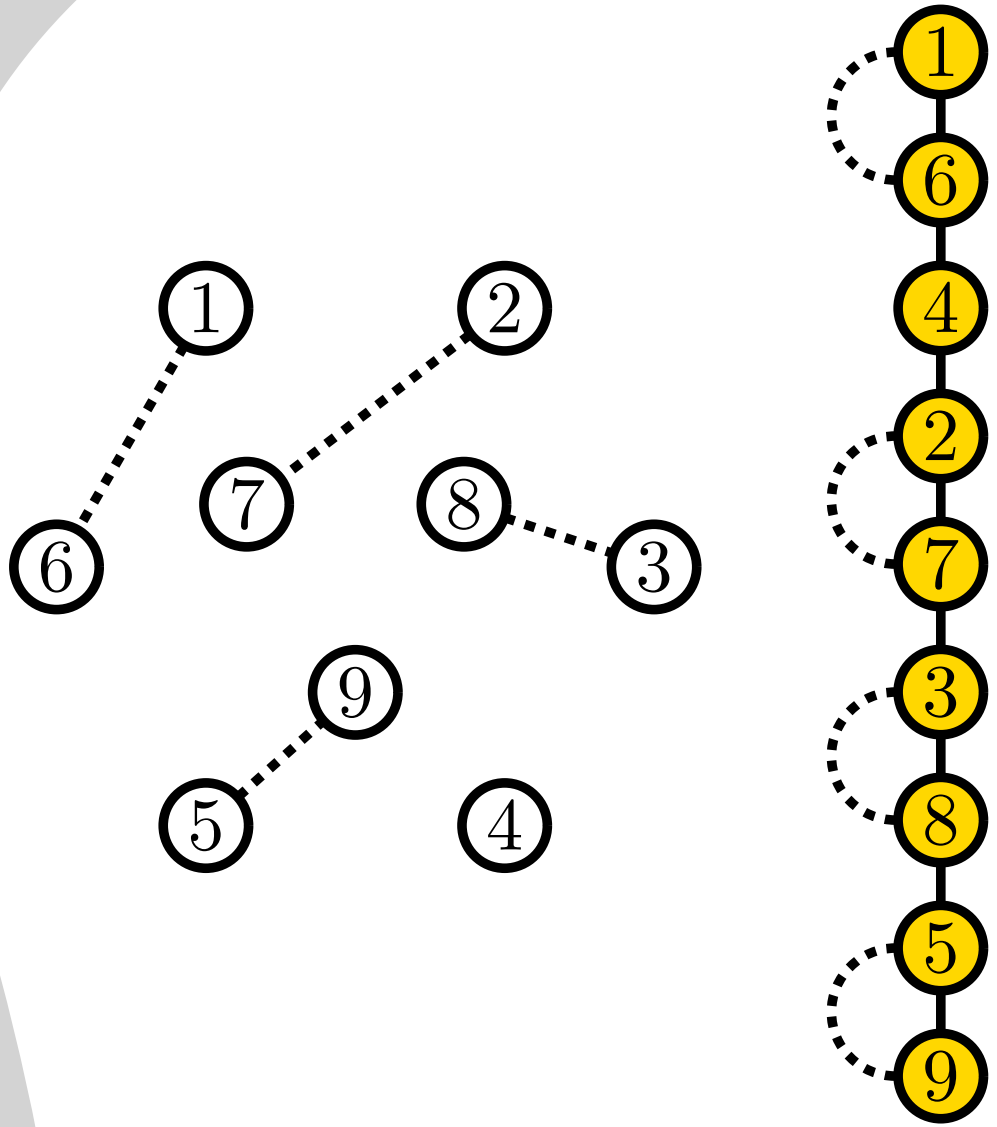




Lower Bound

The Strategy:

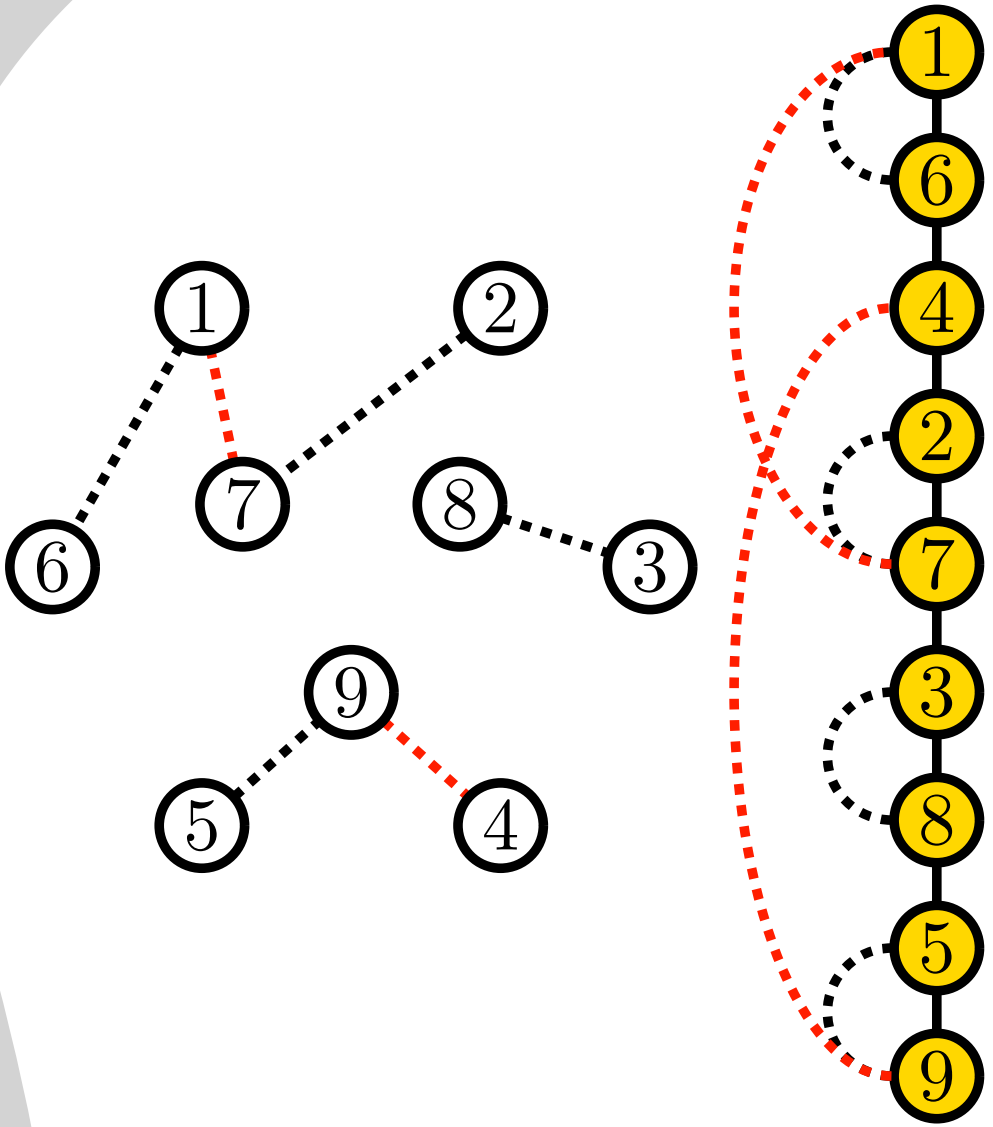
- Exploit Bad Edges
- Introduce Bad Matching



Lower Bound

The Strategy:

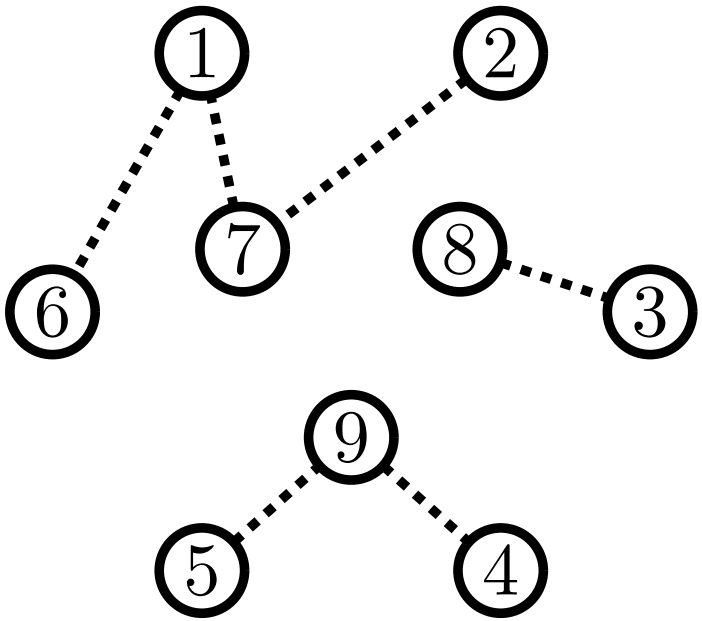
- Exploit Bad Edges
- Introduce Bad Matching



Lower Bound

The Strategy:

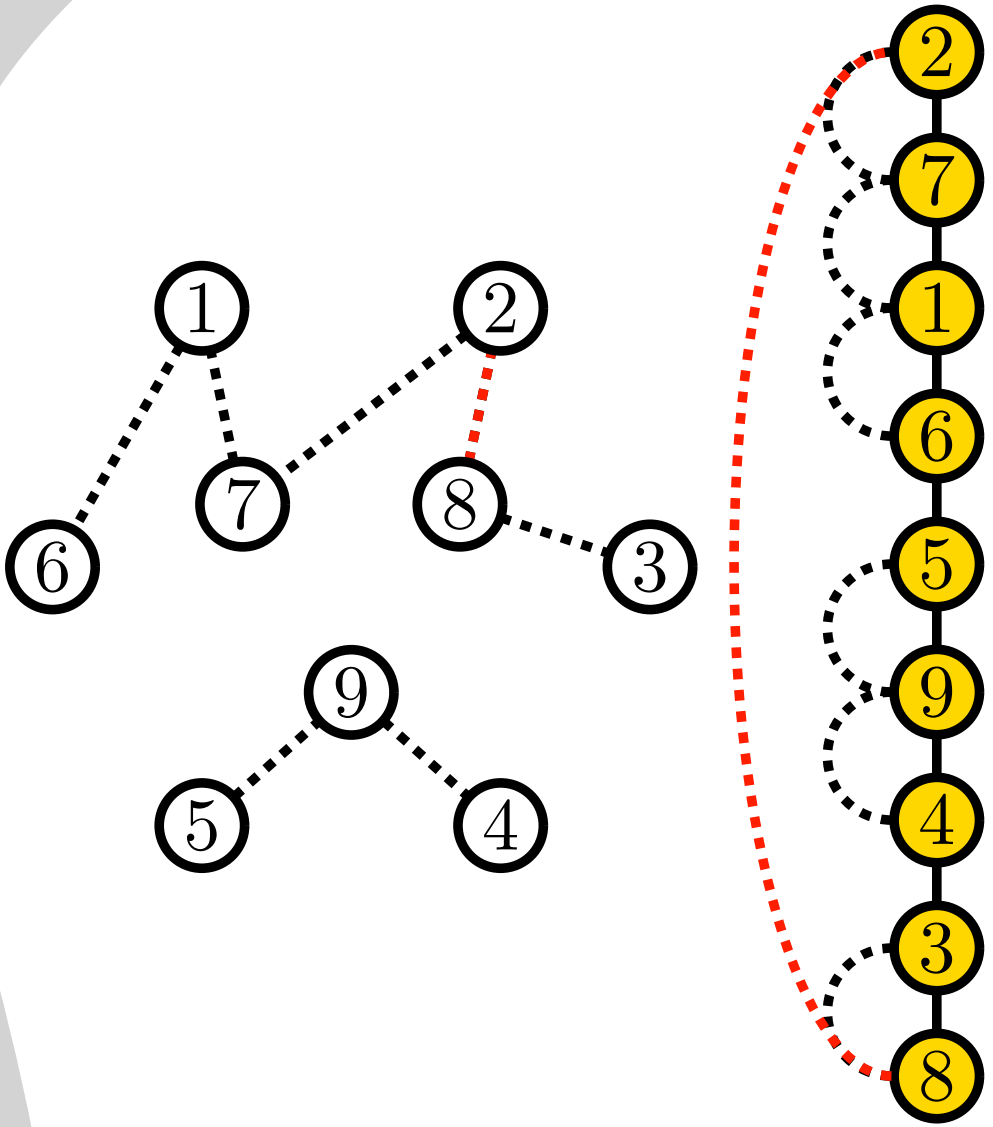
- Exploit Bad Edges
- Introduce Bad Matching



Lower Bound

The Strategy:

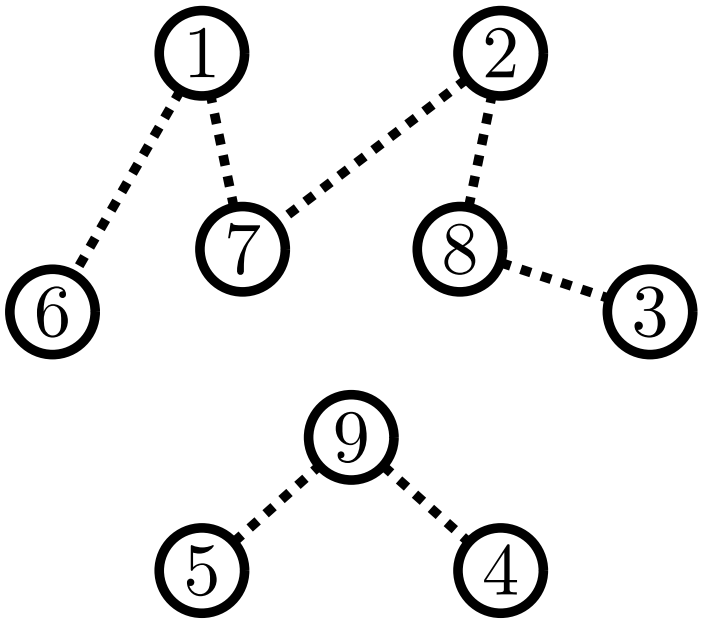
- Exploit Bad Edges
- Introduce Bad Matching

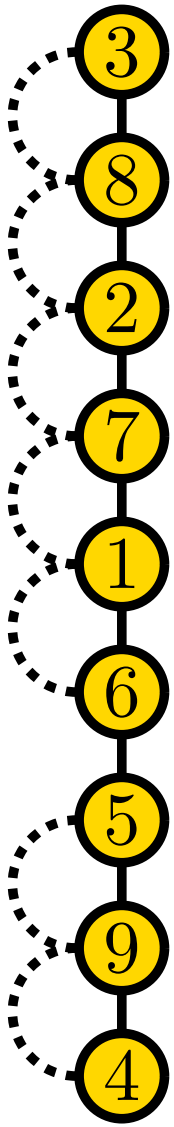


Lower Bound

The Strategy:

- Exploit Bad Edges
- Introduce Bad Matching

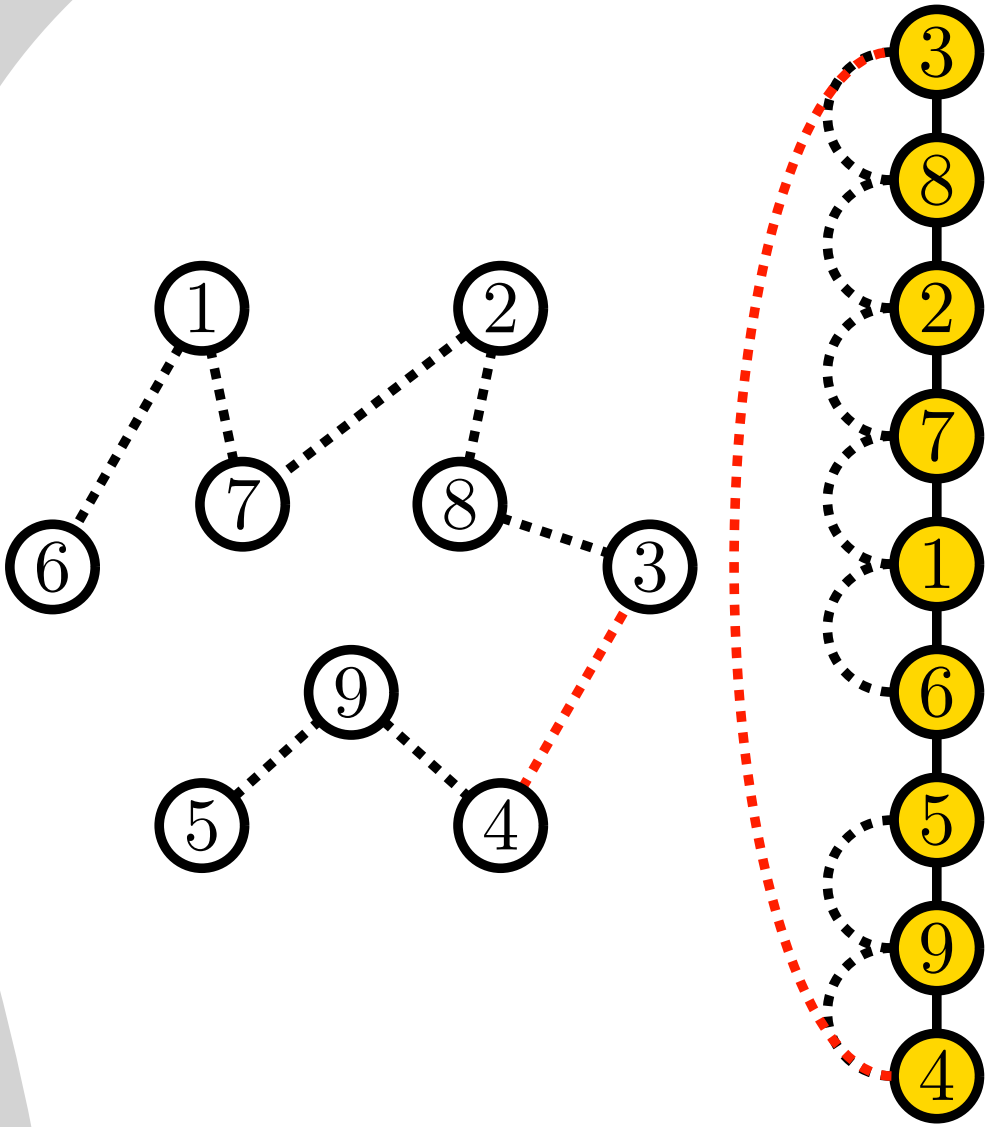




Lower Bound

The Strategy:

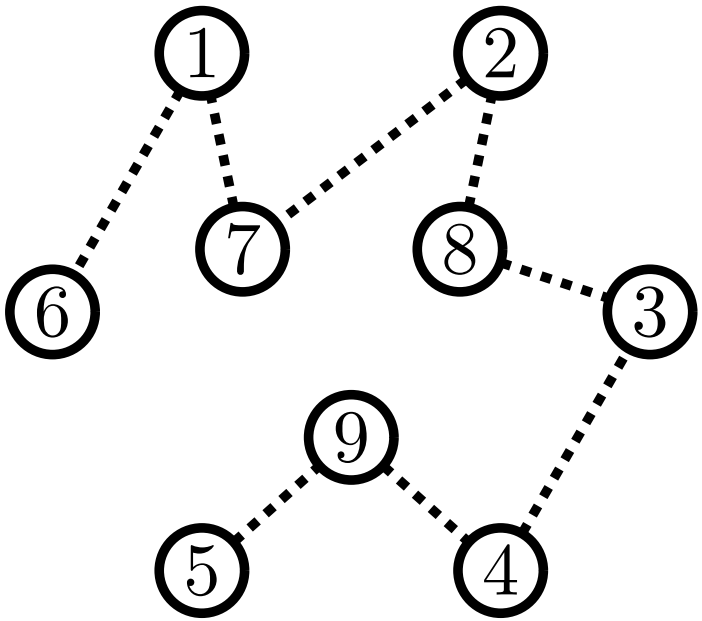
- Exploit Bad Edges
- Introduce Bad Matching

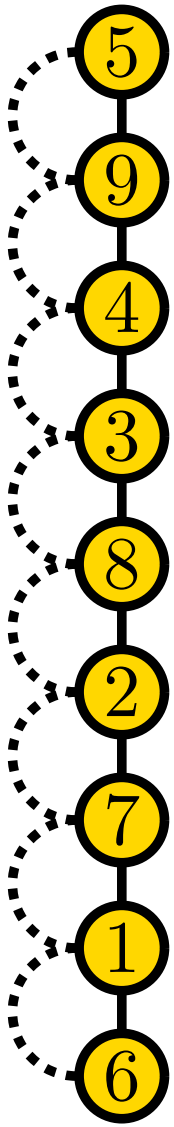


Lower Bound

The Strategy:

- Exploit Bad Edges
- Introduce Bad Matching





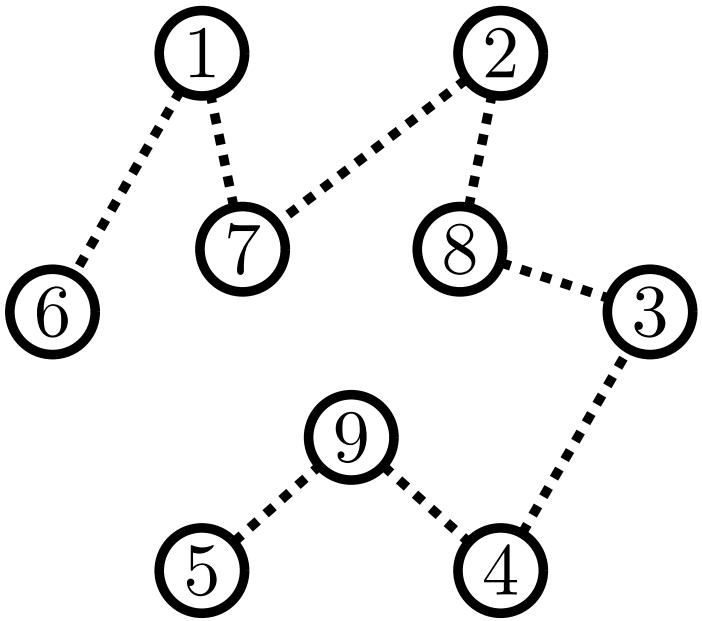
Lower Bound

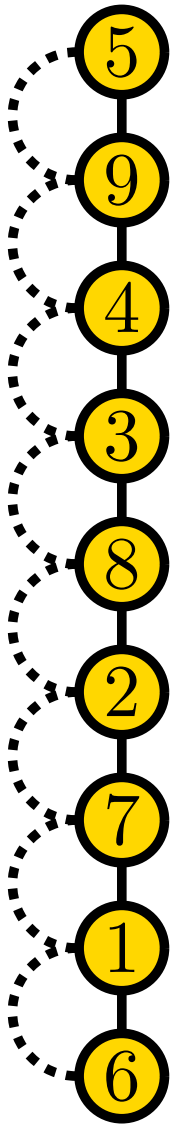
The Strategy:

- Exploit Bad Edges
- Introduce Bad Matching

The Basic Math:

- $\log n$ Bad Matchings
- n^2 Cost per Matching
- Online Cost: $\Omega(n^2 \log n)$





Lower Bound

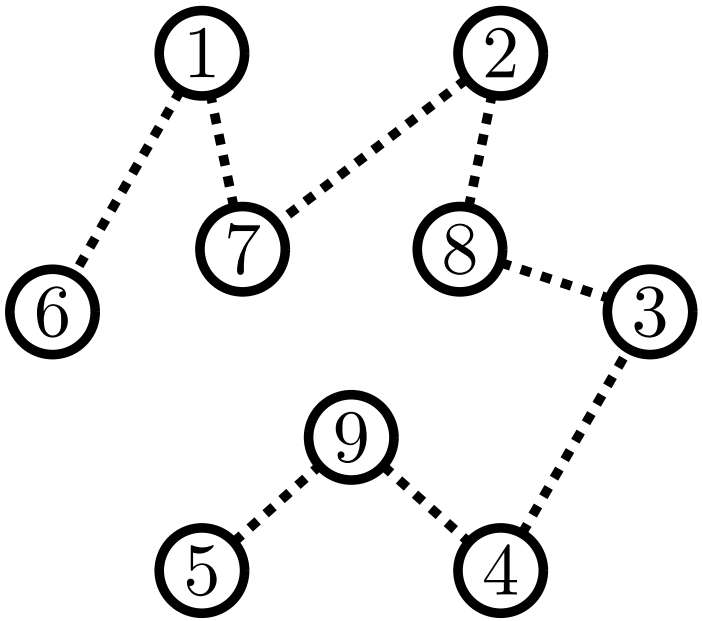
The Strategy:

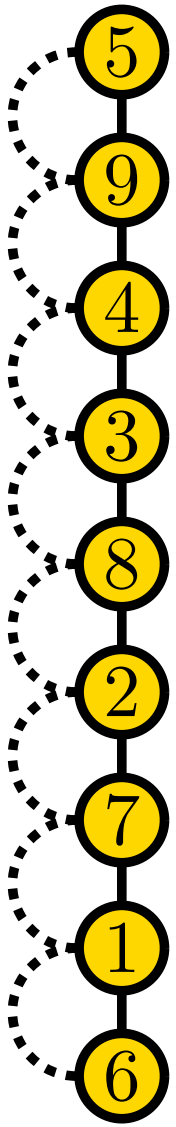
- Exploit Bad Edges
- Introduce Bad Matching
- Maintain List Graph

The Basic Math:

- $\log n$ Bad Matchings
- n^2 Cost per Matching
- Online Cost: $\Omega(n^2 \log n)$
- Offline Cost: $\Theta(n^2)$

  $\rightarrow$ Competitive Ratio: $\Omega(\log n)$





Lower Bound

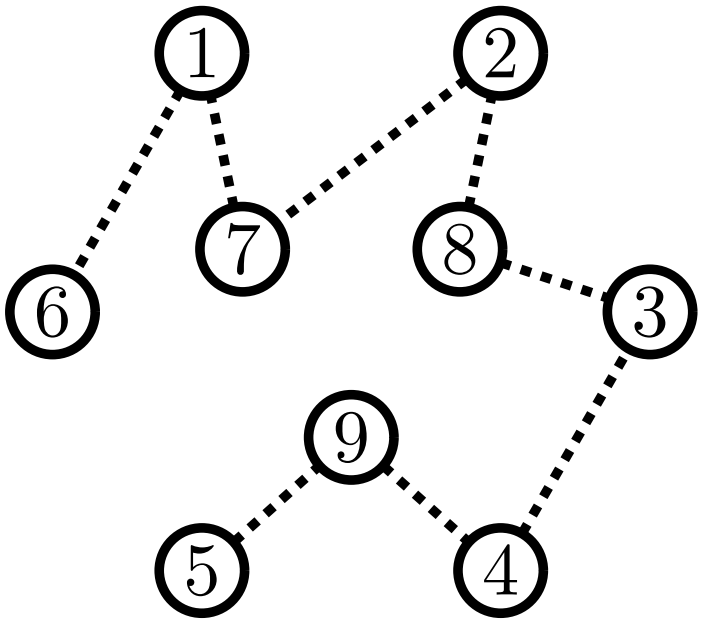
The Strategy:

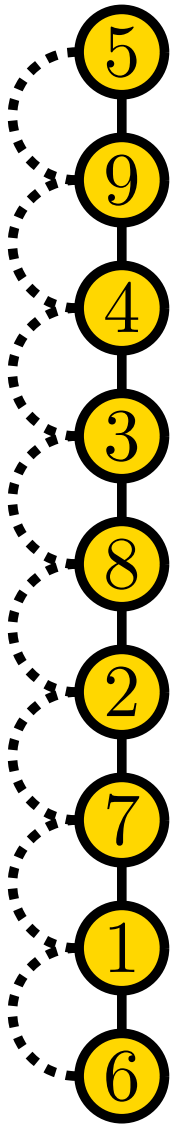
- Exploit Bad Edges
- Introduce Bad Matching
- Maintain List Graph

The Basic Math:

- $\log n$ Bad Matchings
- n^2 Cost per Matching
- Online Cost: $\Omega(n^2 \log n)$
- Offline Cost: $\Theta(n^2)$

  $\rightarrow$ Competitive Ratio: $\Omega(\log n)$





Lower Bound

The Strategy:

- Exploit Bad Edges
- Introduce Bad Matching
- Maintain List Graph

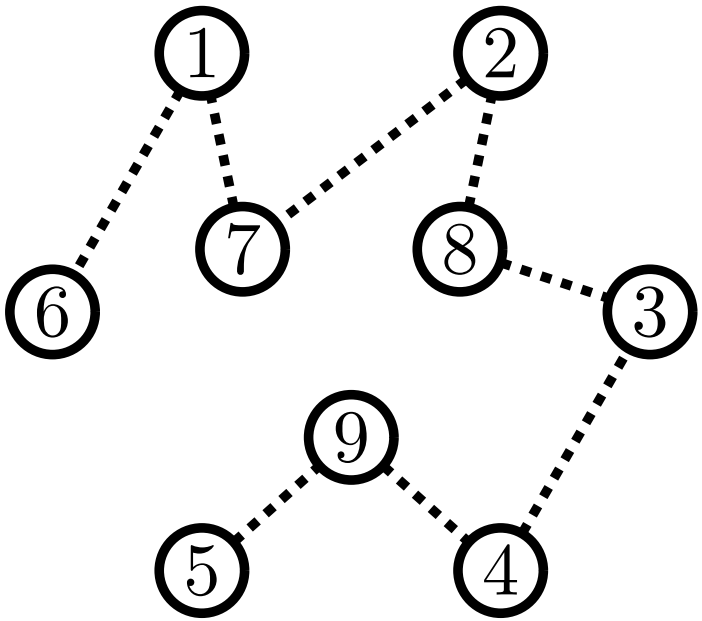
The Basic Math:

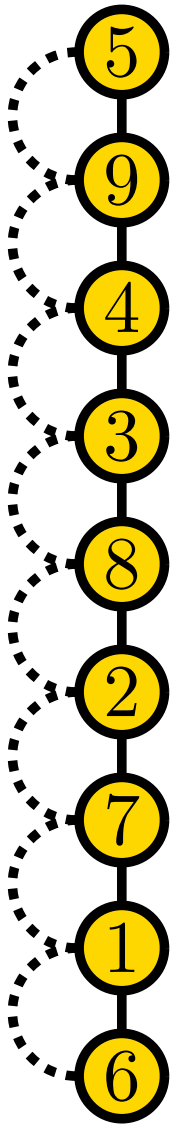
- $\log n$ Bad Matchings
- n^2 Cost per Matching
- Online Cost: $\Omega(n^2 \log n)$
- Offline Cost: $\Theta(n^2)$

  $\rightarrow$ Competitive Ratio: $\Omega(\log n)$

n^2 per Matching:

- Quantify Distortion of Matching





Lower Bound

The Strategy:

- Exploit Bad Edges
- Introduce Bad Matching
- Maintain List Graph

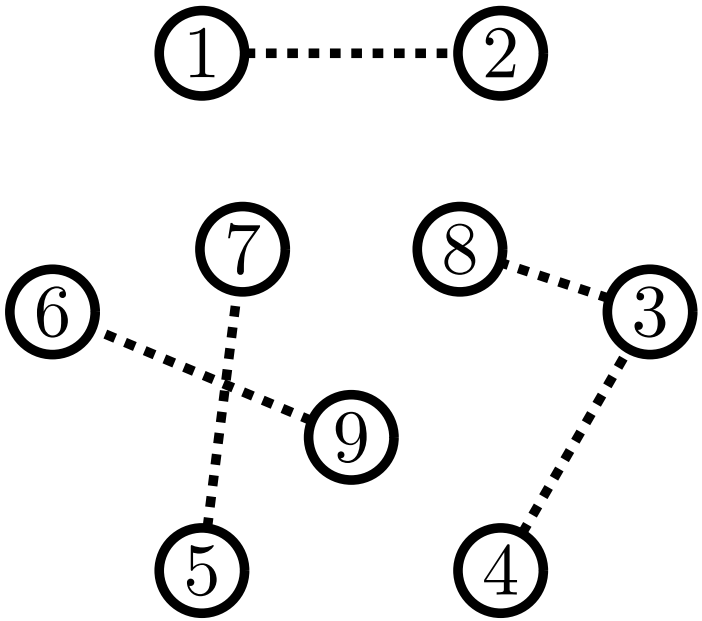
The Basic Math:

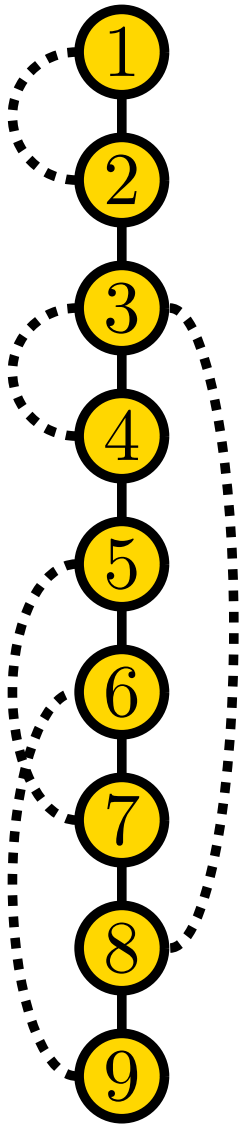
- $\log n$ Bad Matchings
- n^2 Cost per Matching
- Online Cost: $\Omega(n^2 \log n)$
- Offline Cost: $\Theta(n^2)$

  $\rightarrow$ Competitive Ratio: $\Omega(\log n)$

n^2 per Matching:

- Quantify Distortion of Matching





Lower Bound

The Strategy:

- Exploit Bad Edges
- Introduce Bad Matching
- Maintain List Graph

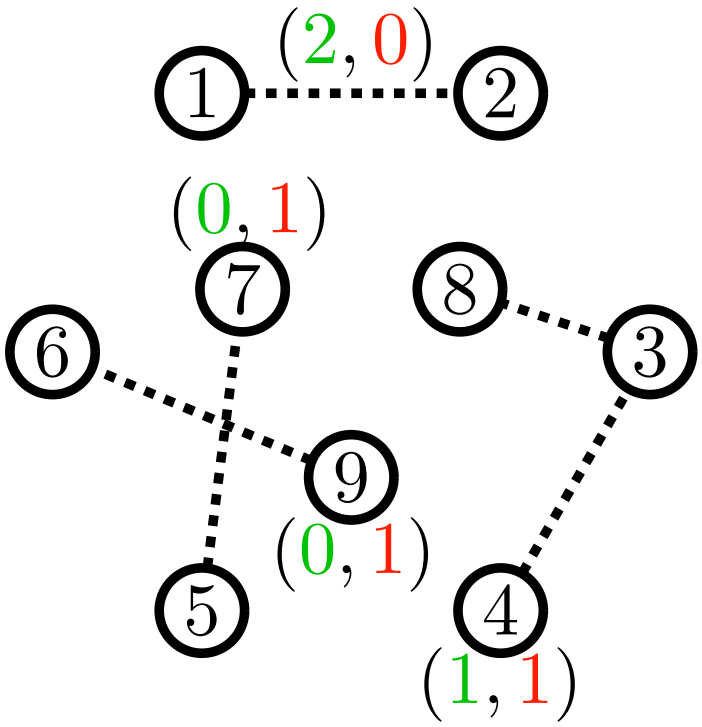
The Basic Math:

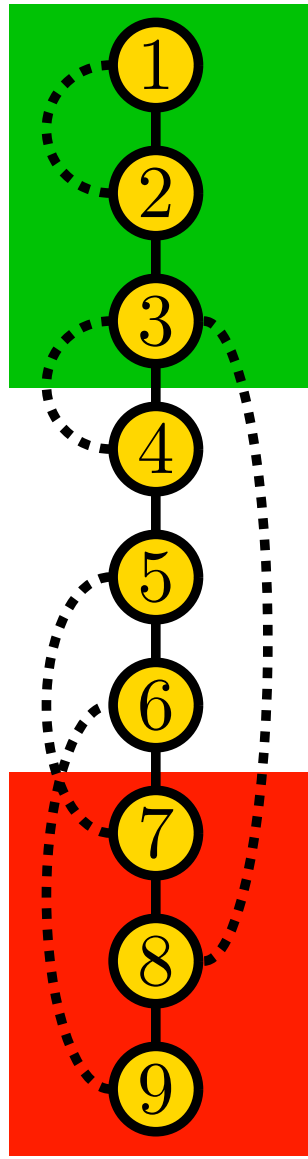
- $\log n$ Bad Matchings
- n^2 Cost per Matching
- Online Cost: $\Omega(n^2 \log n)$
- Offline Cost: $\Theta(n^2)$

  $\rightarrow$ Competitive Ratio: $\Omega(\log n)$

n^2 per Matching:

- Quantify Distortion of Matching





Lower Bound

The Strategy:

- Exploit Bad Edges
- Introduce Bad Matching
- Maintain List Graph

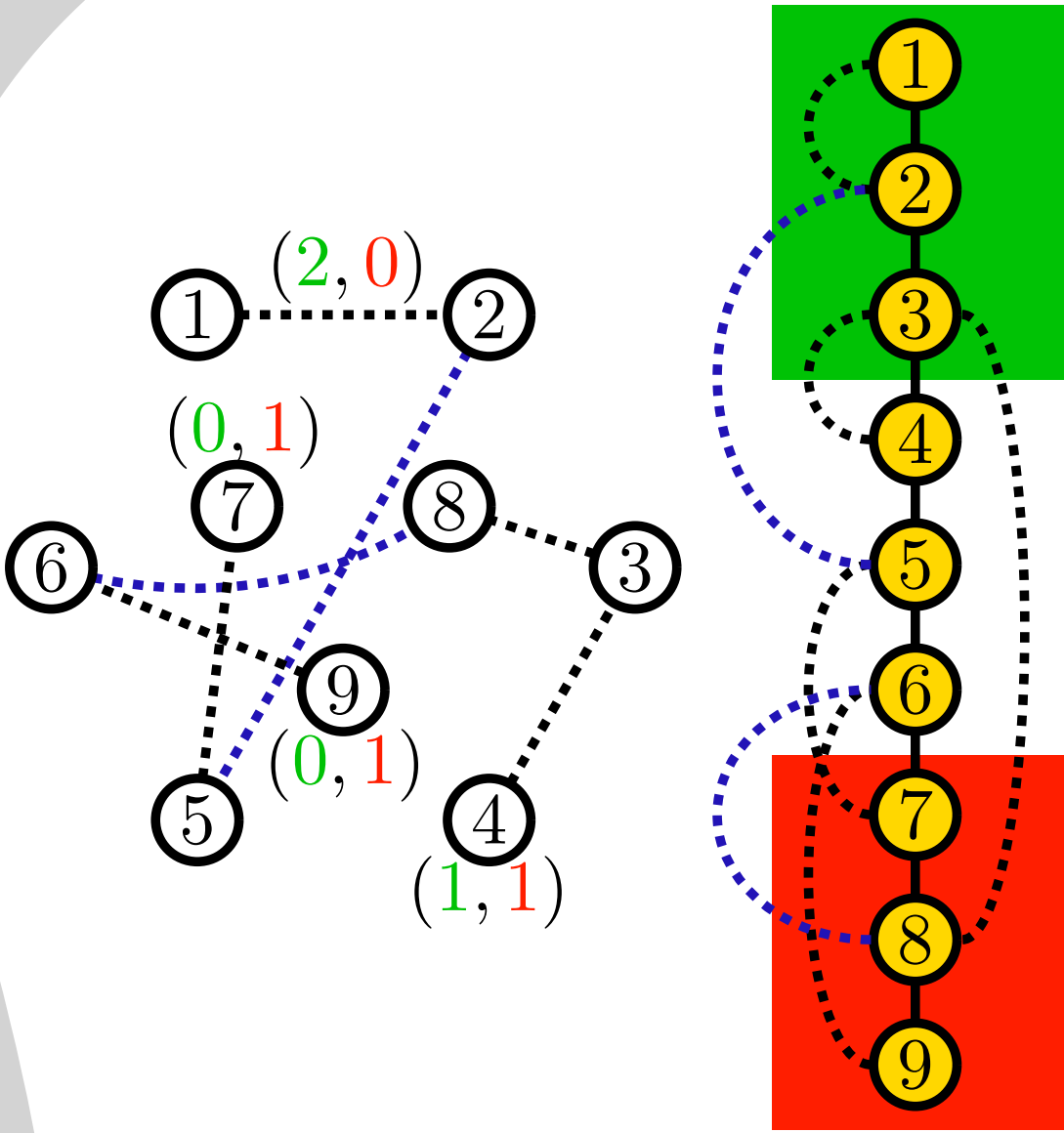
The Basic Math:

- $\log n$ Bad Matchings
- n^2 Cost per Matching
- Online Cost: $\Omega(n^2 \log n)$
- Offline Cost: $\Theta(n^2)$

  $\rightarrow$ Competitive Ratio: $\Omega(\log n)$

n^2 per Matching:

- Quantify Distortion of Matching



Lower Bound

The Strategy:

- Exploit Bad Edges
- Introduce Bad Matching
- Maintain List Graph

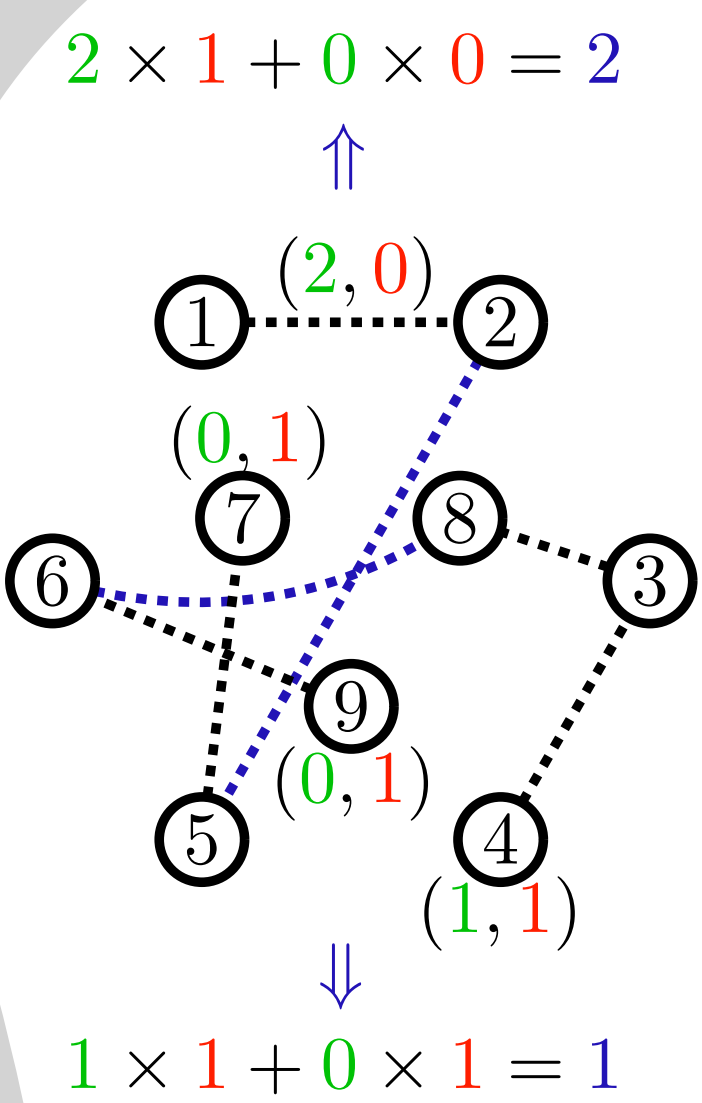
The Basic Math:

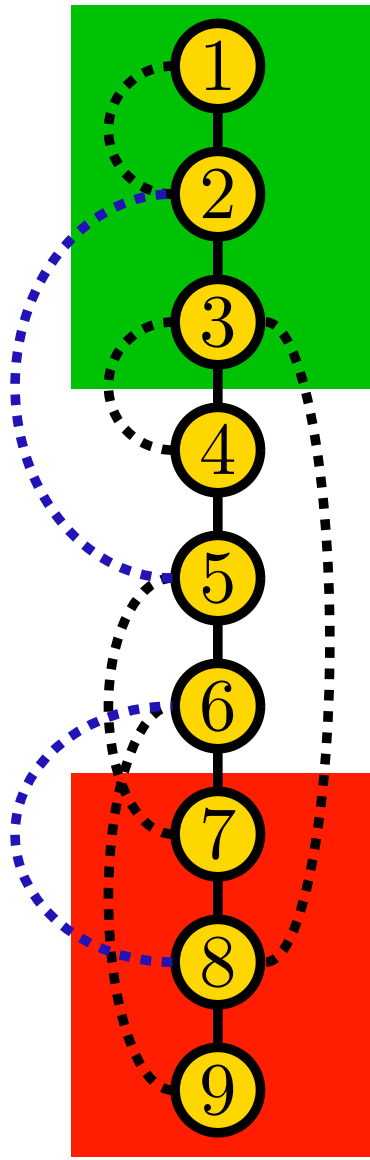
- $\log n$ Bad Matchings
- n^2 Cost per Matching
- Online Cost: $\Omega(n^2 \log n)$
- Offline Cost: $\Theta(n^2)$

  $\rightarrow$ Competitive Ratio: $\Omega(\log n)$

n^2 per Matching:

- Quantify Distortion of Matching





Lower Bound

The Strategy:

- Exploit Bad Edges
- Introduce Bad Matching
- Maintain List Graph

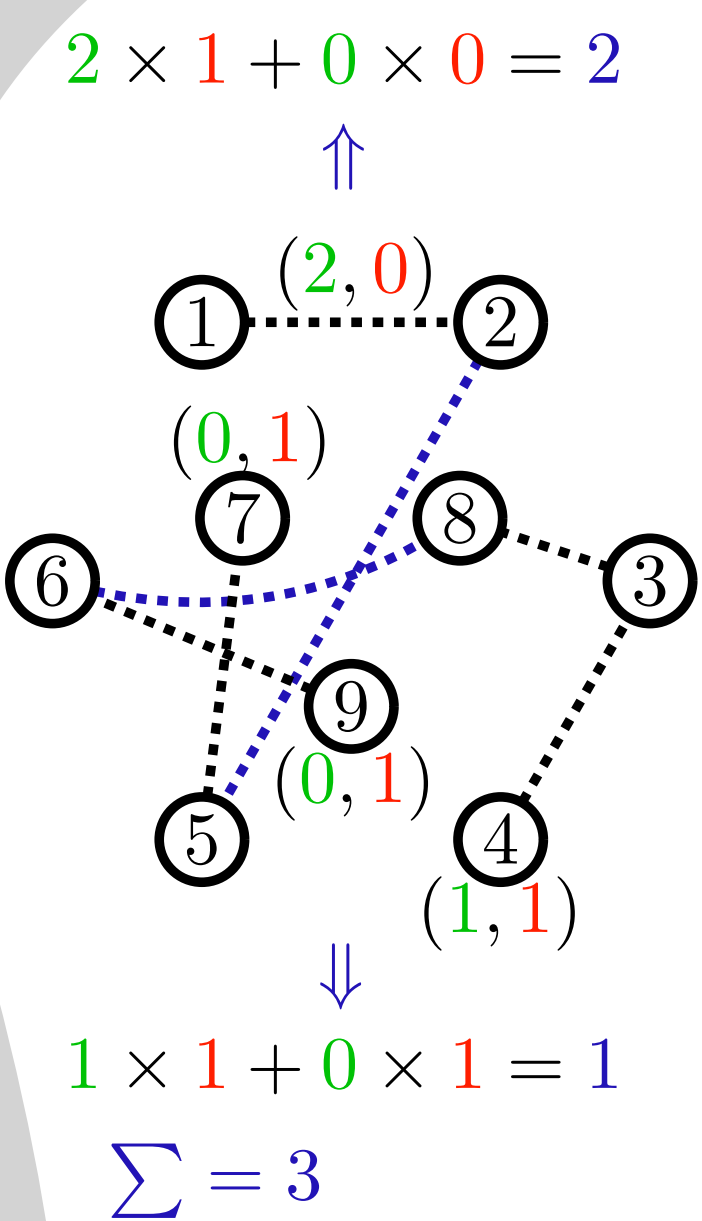
The Basic Math:

- $\log n$ Bad Matchings
- n^2 Cost per Matching
- Online Cost: $\Omega(n^2 \log n)$
- Offline Cost: $\Theta(n^2)$

  $\rightarrow$ Competitive Ratio: $\Omega(\log n)$

n^2 per Matching:

- Quantify Distortion of Matching



Lower Bound

The Strategy:

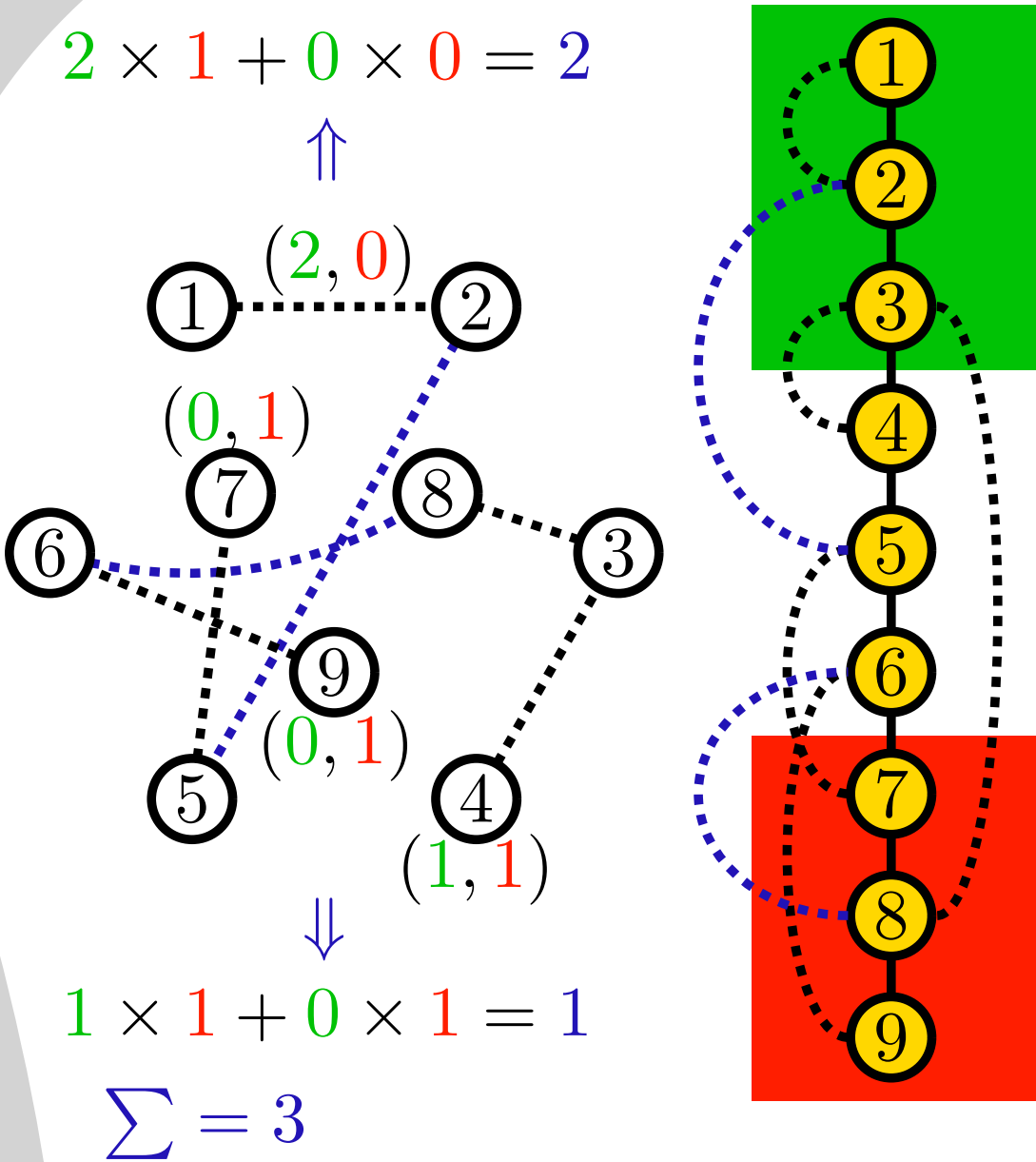
- Exploit Bad Edges
- Introduce Bad Matching
- Maintain List Graph

The Basic Math:

- $\log n$ Bad Matchings
- n^2 Cost per Matching
- Online Cost: $\Omega(n^2 \log n)$
- Offline Cost: $\Theta(n^2)$
  - $\rightarrow$ Competitive Ratio: $\Omega(\log n)$

n^2 per Matching:

- Quantify Distortion of Matching
- Compute Sum of Distortion of all Matchings
- Avereage Distortion is Sufficient

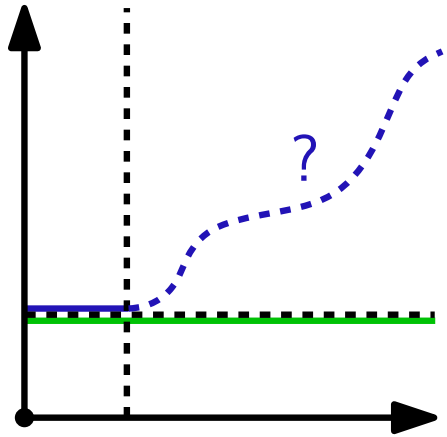


Open Questions & Future Work

Open Questions & Future Work

Better Bounds for Line Networks:

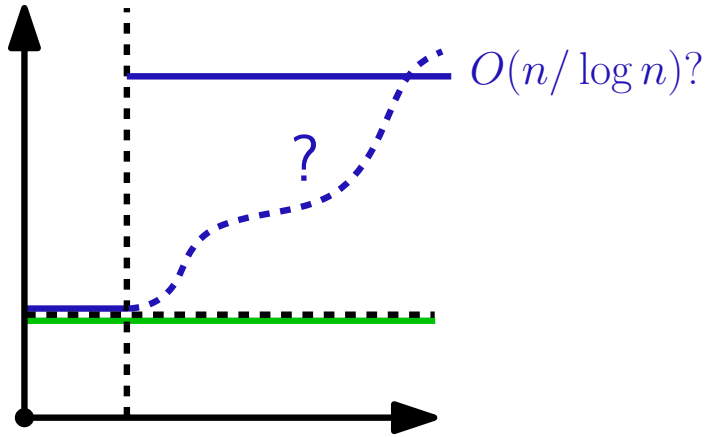
- Nontrivial Upper Bound?
- Better Lower Bound?
- Dynamic Offline Algorithms?



Open Questions & Future Work

Better Bounds for Line Networks:

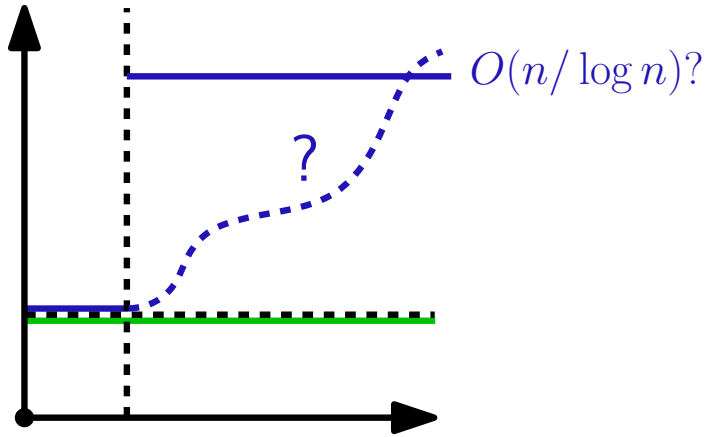
- Nontrivial Upper Bound?
- Better Lower Bound?
- Dynamic Offline Algorithms?



Open Questions & Future Work

Better Bounds for Line Networks:

- Nontrivial Upper Bound?
- Better Lower Bound?
- Dynamic Offline Algorithms?



Different Networks:

- Binary (Splay) Trees
- No BST Property