



INSTITUT  
DE RECHERCHE  
EN INFORMATIQUE  
FONDAMENTALE

# Distributed and Non-Distributed Computational Models

Ami Paz

IRIF – CNRS and Paris Diderot University

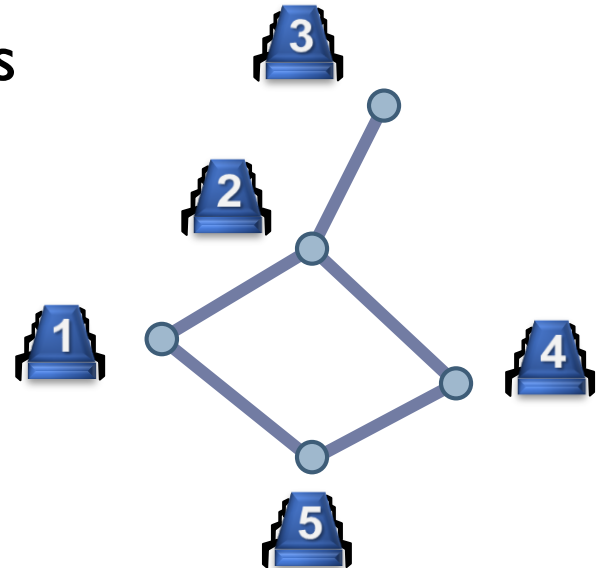
# Message Passing Models

1. **LOCAL**
2. **CONGEST**
3. **CLIQUE**

# Message Passing Models

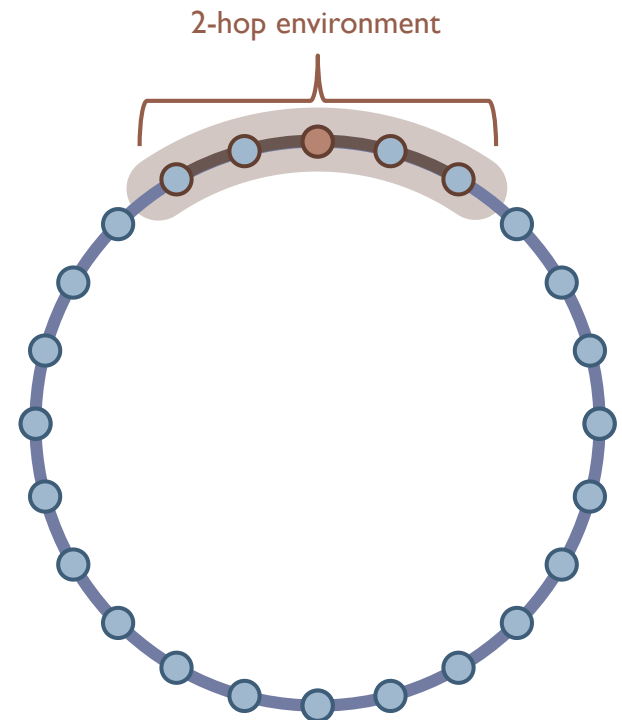
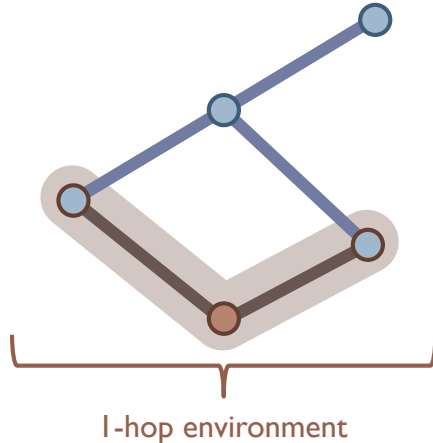
---

- ▶ A graph  $G = (V, E)$  representing the network's topology
- ▶  $n$  unbounded processors, located on the nodes
- ▶ Communicating on the edges
- ▶ Synchronous network
- ▶ Compute / verify graph parameters



# The LOCAL Model

- ▶ Unbounded messages
- ▶ Solving local tasks:
  - ▶ Coloring
  - ▶ MST
  - ▶ MIS
- ▶ Anything\* solvable in  $O(D)$  rounds



# Two Examples

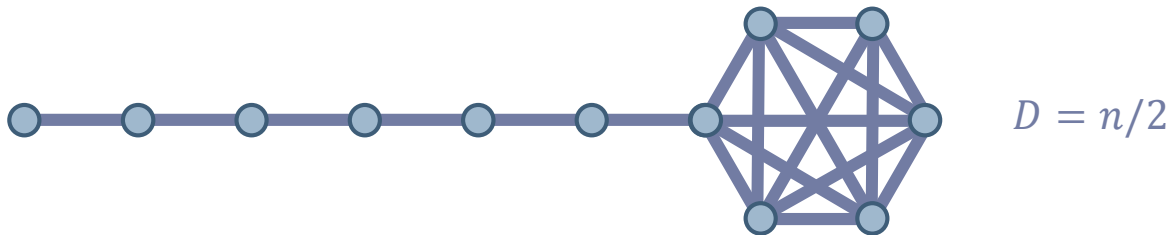
---

- ▶ Triangle detection
  - ▶ Easy, in one round
  - ▶ Send all your neighbors your list of neighbors
  
- ▶ Computing the diameter  $D$ 
  - ▶ Takes  $\Theta(D)$  rounds

# Diameter Lower Bound

---

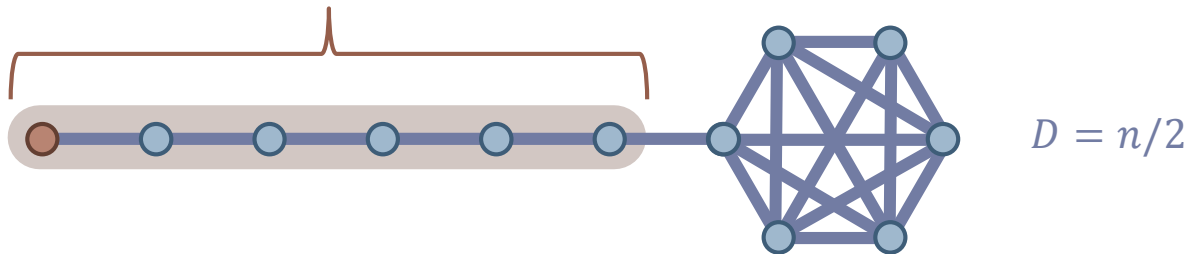
- ▶ Computing  $D$  takes  $\Omega(D)$  rounds
  - ▶ Indistinguishability argument



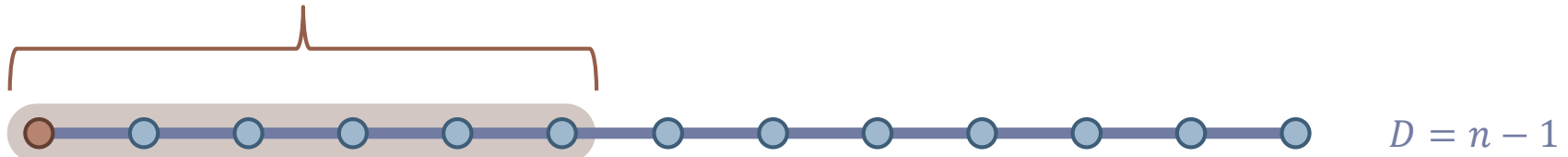
# Diameter Lower Bound

- ▶ Computing  $D$  takes  $\Omega(D)$  rounds
  - ▶ Indistinguishability argument

View after  $n/2 - 1$  rounds



View after  $n/2 - 1$  rounds

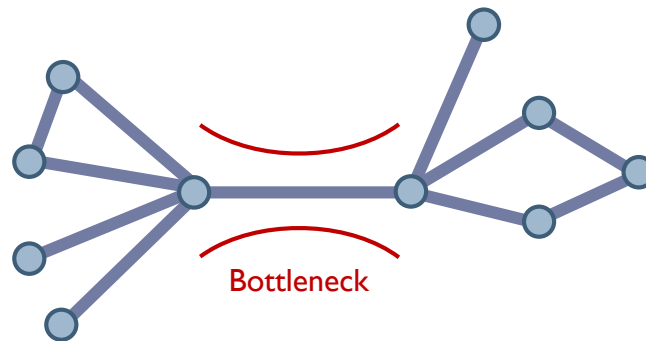


Cannot distinguish

# The **CONGEST** Model

---

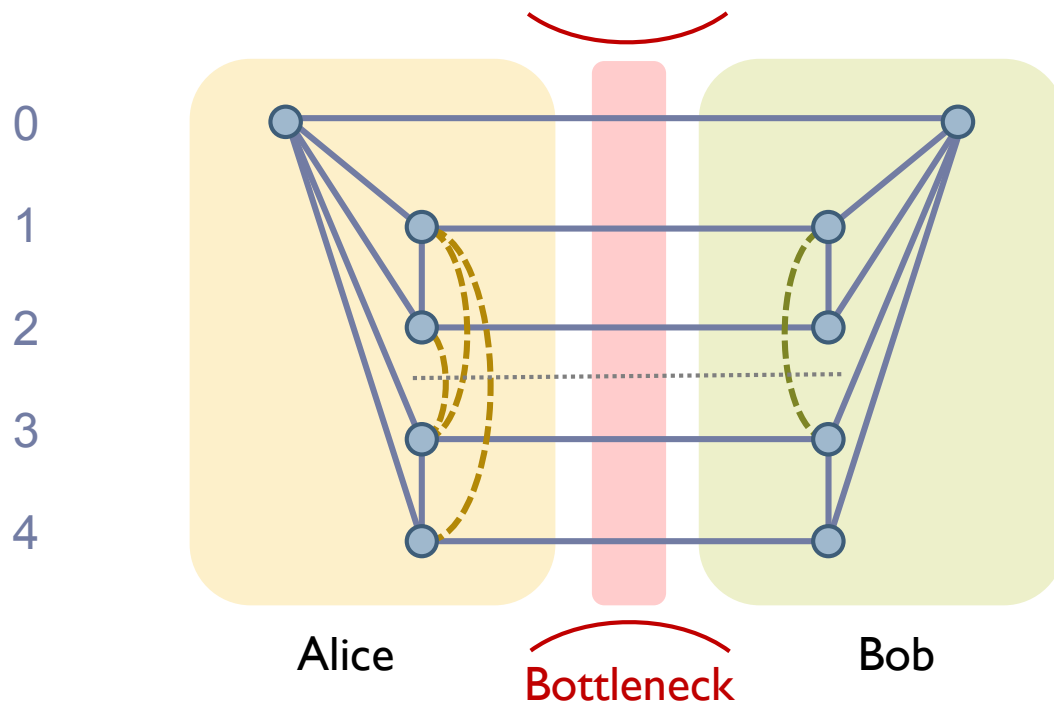
- ▶ Bounded message size; typically  $b = O(\log n)$
- ▶ All **LOCAL** lower bounds still hold
- ▶ Some **LOCAL** algorithms still work
  - ▶ But not all!





# CONGEST – Typical Lower Bound [HW12]

- ▶ Communication complexity problem
- ▶ Inputs encoded by a graph
- ▶ Split the graph between Alice and Bob
- ▶ CC lower bounds imply message lower bounds

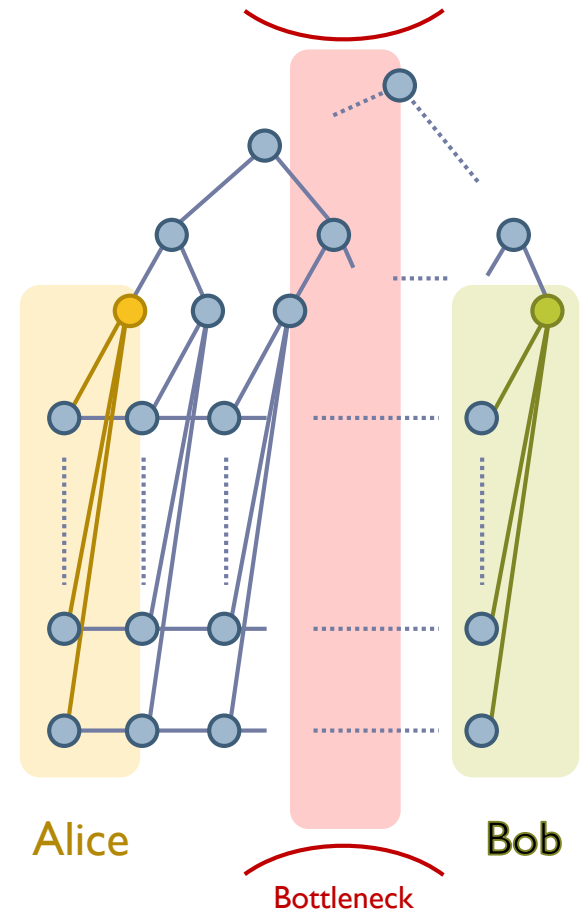
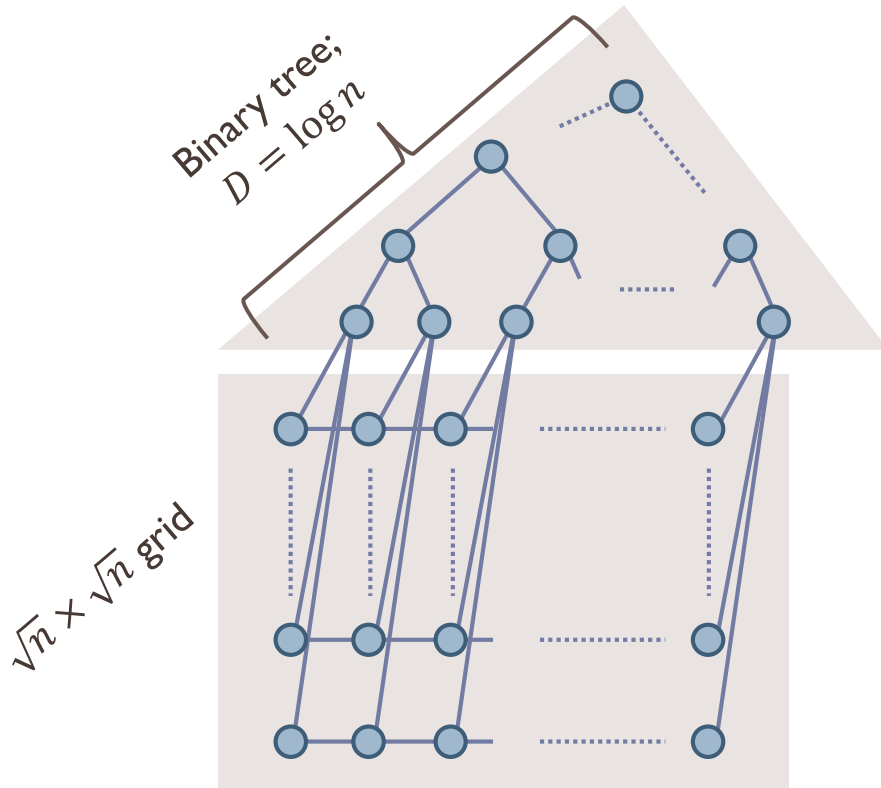


Disjointness on  
 $\Theta(n^2)$  bits.

Diam 2 or 3?

- Diam 2 – disjoint
  - Diam 3 – not disjoint
- $\tilde{\Omega}(n)$  rounds are needed

# CONGEST – Another Lower Bound



$\Omega(\sqrt{n}/b)$  lower bound

Verification: MST, bipartiteness, cycle, connectivity...

Approximation: MST, min cut, shortest s-t path...

Verifying MST is harder  
than finding one!

# So Far:

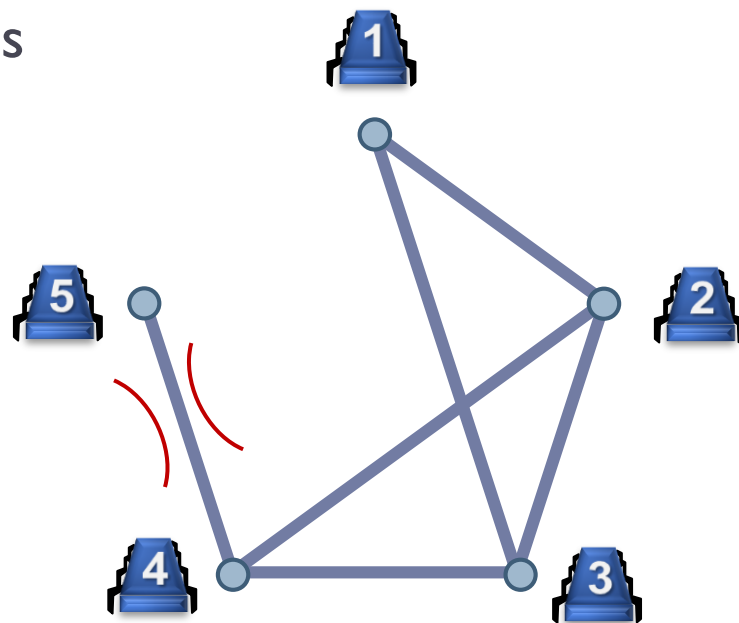
---

- ▶ **LOCAL** model:

- ▶ Unbounded messages
- ▶ Everything is solvable in  $O(D)$  rounds

- ▶ **CONGEST** model:

- ▶ Message =  $O(\log n)$  bits
- ▶ Lower bounds of  $\tilde{\Omega}(\sqrt{n} + D)$
- ▶ Tight for many problems

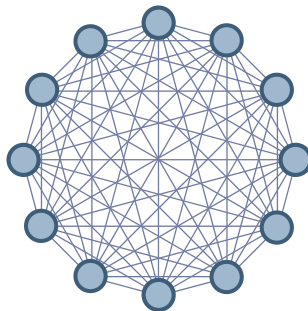


- ▶ Question: is  $\Omega(\sqrt{n})$  due to congestion?

# The **CLIQUE** Model

---

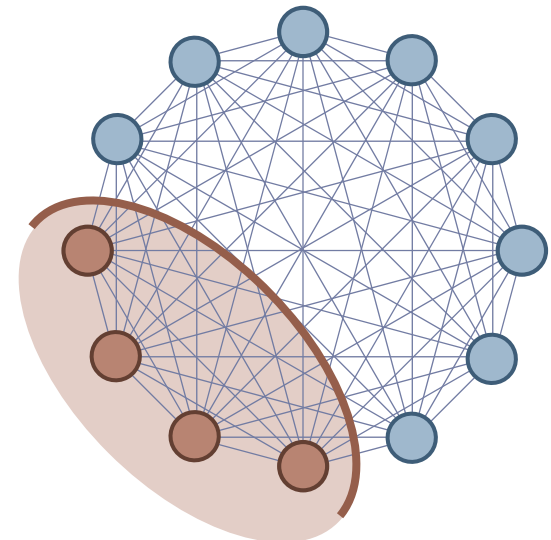
- ▶ All-to-all message passing – a clique network
- ▶ Diameter of 1
- ▶ No distance – only congestion
  
- ▶ MST in  $O(\log^* n)$  rounds [GPI6]
  - ▶ Fast triangle detection, diameter, APSP, ...



# CLIQUE – Lower Bound?

---

- ▶ Diam = 1
- ▶ Larger set – more outgoing edges
- ▶ No nontrivial lower bound is known
- ▶ Simple counting argument [DKO14]
  - ▶ many functions need  $n - 5 \log n$  rounds



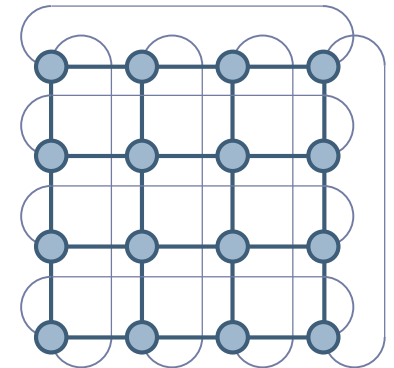
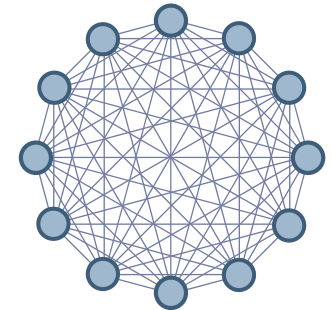
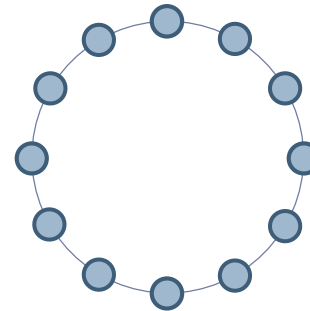


# Parallel Systems

# Parallel Systems

---

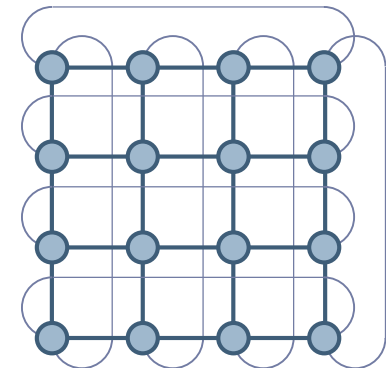
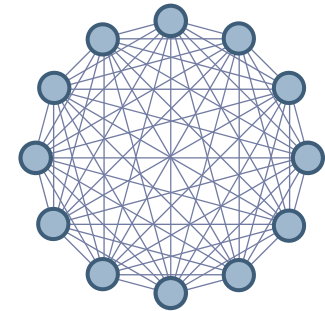
- ▶  $n$  synchronous processors,  $k$  inputs to each
- ▶ Connected by a communication graph
- ▶ Typical graphs:
  - ▶ Clique
  - ▶ Cycle
  - ▶ Torus (Grid)
- ▶ Known topology, known identities
- ▶ Bounded message size
- ▶ Bounded memory
- ▶ Bounded computational power



# Parallel vs. CONGEST

---

- ▶ **Parallel is more restrictive:**
  - ▶ Bounded memory
  - ▶ Bounded computational power
- ▶ **Different focus:**
  - ▶ Specific communication graphs
  - ▶ Algebraic questions vs. graph parameters





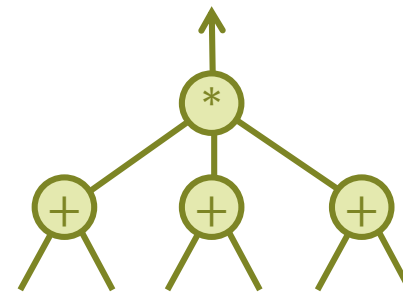
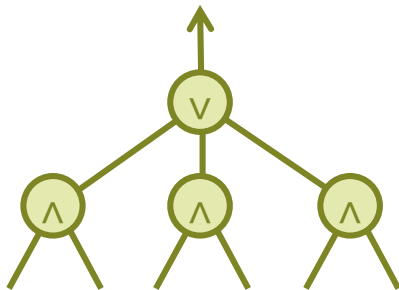


# Circuits

# Circuits

---

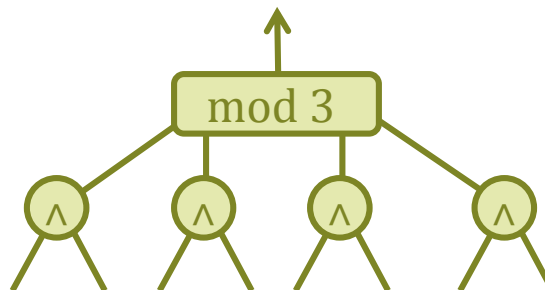
- ▶ Algebraic computation model
- ▶ A computation graph (circuit) composed of:
  - ▶ Inputs, output, and operation gates
- ▶ Represent many algorithms:
  - ▶ Matrix multiplication, determinant, permanent
- ▶ Complexity measures:
  - ▶ Depth, number of gates, fan-in, fan-out



# Circuits Families

---

- ▶ Arithmetic circuits
- ▶ Boolean circuits
- ▶ Boolean circuits augmented with:
  - ▶ mod  $m$  gates
  - ▶ Threshold gates
  - ▶ ...



# Circuits Lower Bounds

---

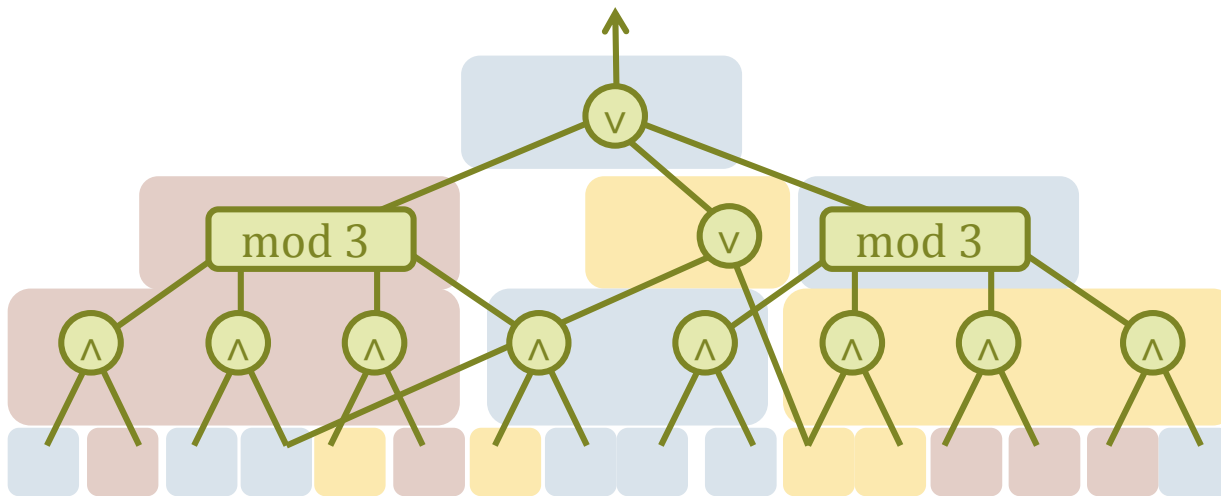
- ▶ What can be computed in constant depth?
- ▶ Counting argument:
  - ▶ Many functions cannot be computed using Boolean circuits
  - ▶ ... or even using augmented circuits
- ▶ But:
  - ▶ No explicit function is known

Circuits  $\Leftrightarrow$  CLIQUE

# CLIQUE vs. Circuits

---

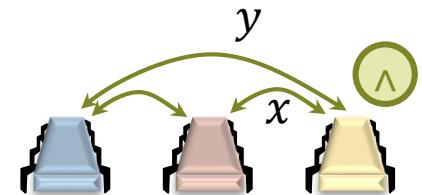
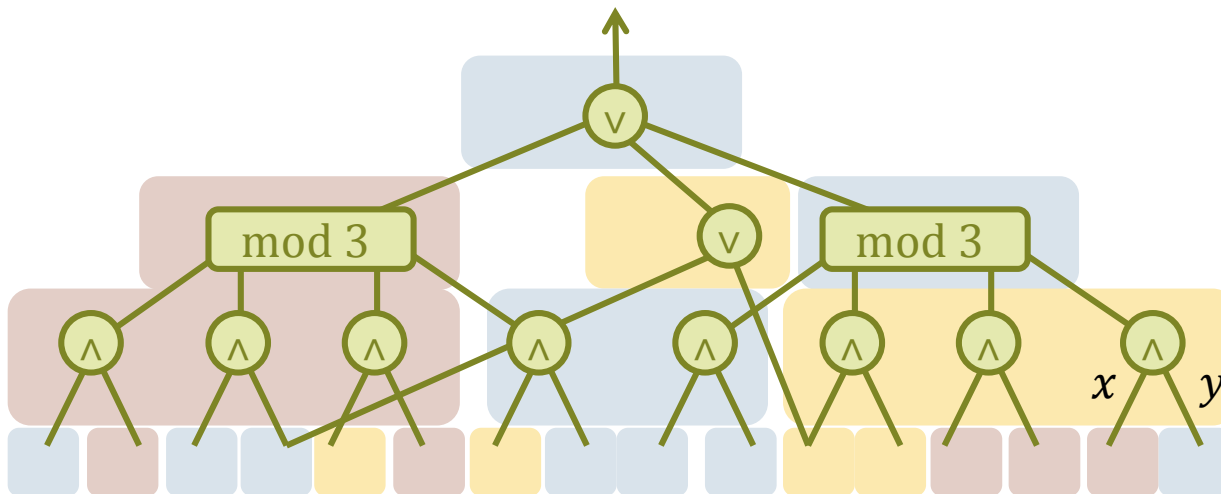
- ▶ **CLIQUE** can simulate circuits [DKO14]
  - ▶ Each node simulates a set of gates in a layer
  - ▶ Circuit's depth = # of rounds



# CLIQUE vs. Circuits

- ▶ Main idea:

- ▶ Simulate each layer of the circuit in  $O(1)$  rounds

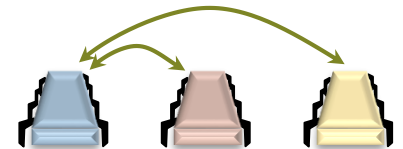
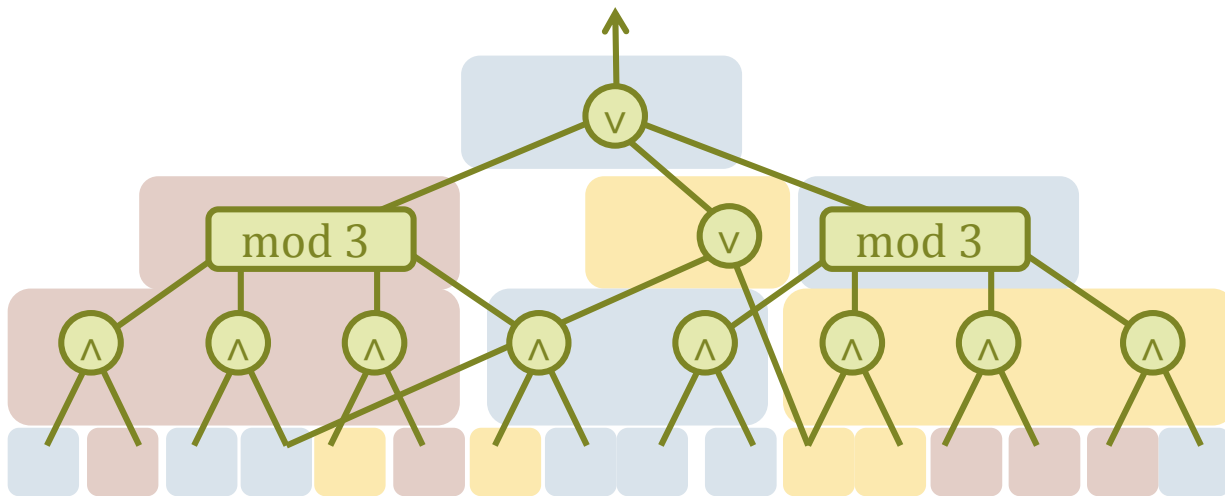


# CLIQUE vs. Circuits

---

- ▶ Main idea:

- ▶ Simulate each layer of the circuit in  $O(1)$  rounds



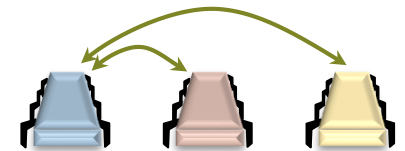
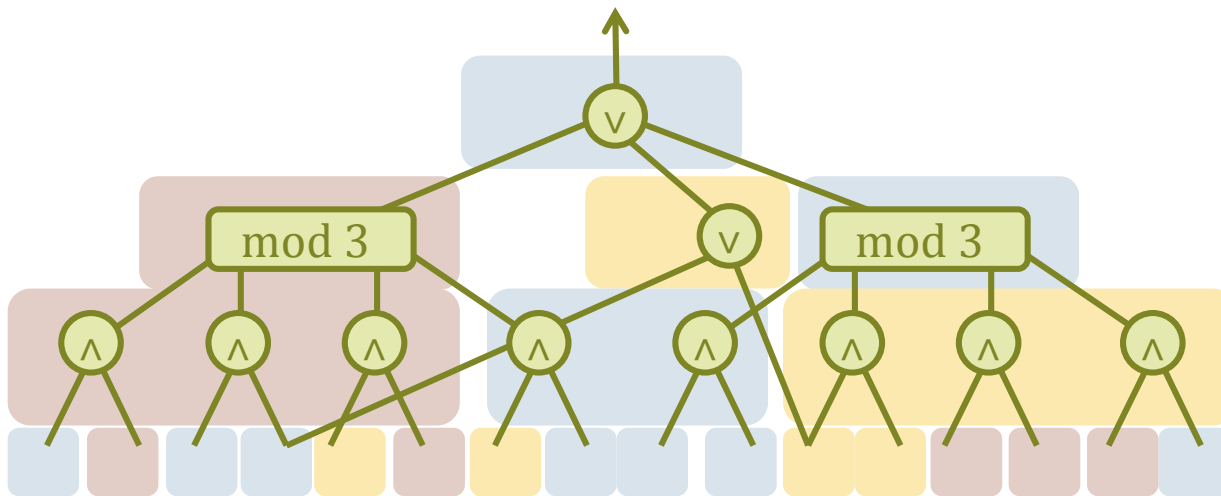


# CLIQUE vs. Circuits

---

- ▶ Main idea:

- ▶ Simulate each layer of the circuit in  $O(1)$  rounds

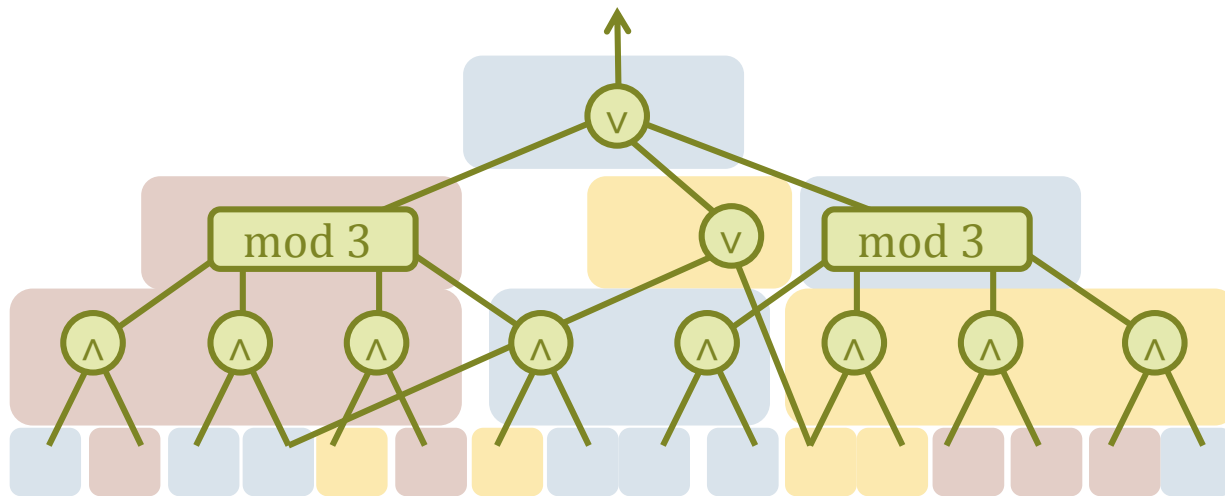


# CLIQUE vs. Circuits

---

- ▶ Main idea:

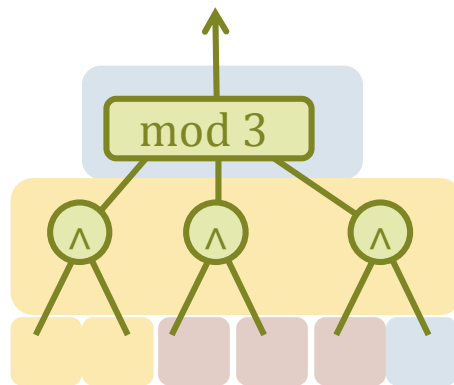
- ▶ Simulate each layer of the circuit in  $O(1)$  rounds



# CLIQUE vs. Circuits

---

- ▶ **CLIQUE** can simulate circuits
  - ▶ Non-constant **rounds** lower bound for the **CLIQUE**  $\Rightarrow$   
Non-constant **depth** lower bound for **circuits**
- ▶ There is also a reduction in the other direction [DKO14]
  - ▶ A circuit can simulate the **CLIQUE**

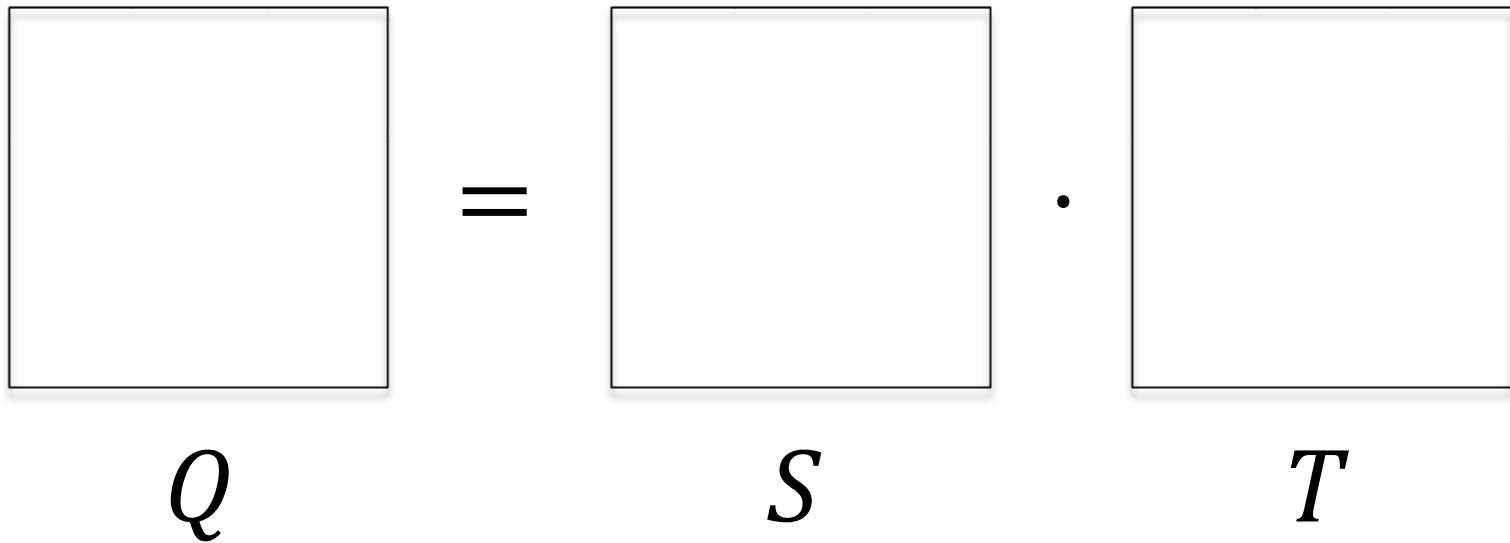


Parallel  $\Leftrightarrow$  CLIQUE

# Matrix Multiplication

---

- ▶ Base for many algebraic problems
- ▶ Thoroughly studied in parallel computing
- ▶ Several algorithms:
  - ▶ different topologies, input / output partitions


$$Q = S \cdot T$$

# Matrix Multiplication

---

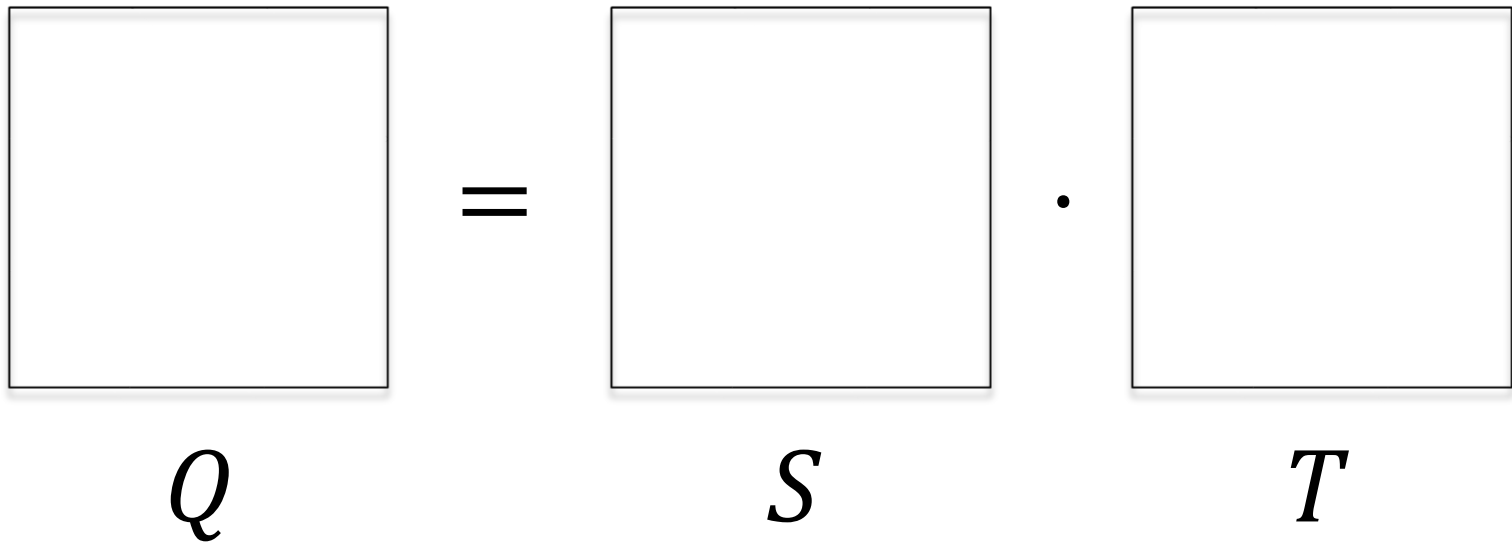
- ▶ This talk:
  - ▶ The 3D algorithm [ABG+95]
  - ▶ For  $n \times n$  matrices and  $n$  processors
  - ▶ Adaptation of parallel algorithm to the **CLIQUE** [CHK+16]

$$Q = S \cdot T$$

# Matrix Multiplication

---

- ▶ Parallel 3D algorithm  $\Rightarrow$
- ▶ **CLIQUE** matrix multiplication in  $O(n^{1/3})$  rounds
  - ▶ Implies triangle detection,  $D$ , APSP, ...
  - ▶ In similar time [CHK+16]



A diagram illustrating matrix multiplication. It consists of three empty square boxes arranged horizontally. The first box is labeled with the letter  $Q$  below it. To its right is an equals sign (=). To the right of the equals sign is a second empty square box labeled with the letter  $S$  below it. To the right of the second box is a dot (·). To the right of the dot is a third empty square box labeled with the letter  $T$  below it. This represents the equation  $Q = S \cdot T$ .

# Fast Matrix Multiplication

---

- ▶ **Standard matrix multiplication:**
  - ▶ Compute  $n^2$  entries, each need  $n$  multiplications
  - ▶ Total:  $\Theta(n^3)$  time
- ▶ **There exist faster algorithms:**
  - ▶ Strassen  $O(n^{2.807})$  [1969]
  - ▶ Coopersmith-Vinograd  $O(n^{2.376})$  [1990]
  - ▶ ...
  - ▶ Le Gall  $O(n^{2.373})$  [2014]
- ▶ **Can be implemented in the CLIQUE**
  - ▶ Distributed matrix multiplication in  $O(n^{0.158})$  rounds



## Some Results & Conclusion

# Triangle Detection in the CLIQUE

---

## 1. Combinatorial algorithm:

- ▶  $O(n^{1/3})$  rounds [DLPI2]

## 2. Reduction from circuits for matrix multiplication:

- ▶  $(n^{\omega-2}) \approx O(n^{0.373})$  rounds, randomized [DKO14]

## 3. Using a technique from parallel matrix multiplication:

- ▶  $O(n^{1-2/\omega}) \approx O(n^{0.158})$  rounds [CHK+16]

## ▶ 2,3 Imply similar complexities for:

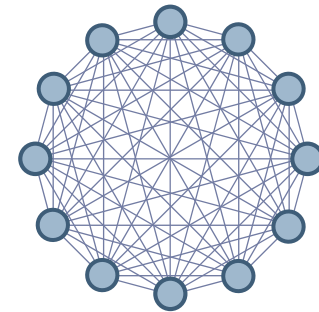
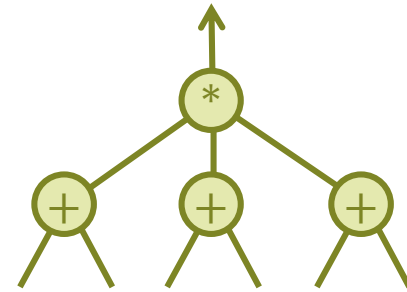
- ▶ APSP, diameter, girth

Sequential matrix multiplication:  
 $O(n^\omega)$  operations

# Conclusion

---

- ▶ **Several models:**
  - ▶ Message passing
    - ▶ **LOCAL, CONGEST** and **CLIQUE**
  - ▶ Parallel systems
  - ▶ Circuits
    - ▶ Arithmetic, Boolean, augmented
- ▶ Many connections and similarities
- ▶ Approach different questions
- ▶ Using different techniques



*Thank You!*