Majority Circuits and Sorting Networks of Small Depth

Vladimir Podolskii

joint Natalia Dobrokhotova-Maikova,
Alexander Kozachinskiy and Alexander Kulikov

WDCM-2022

Majority

Boolean functions:

$$f\colon \{0,1\}^n \to \{0,1\}$$

Standard majority:

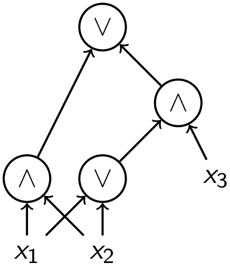
$$\mathrm{MAJ}_n(x_1,\ldots,x_n) = 1 \iff \sum_i x_i > n/2$$

More general version:

$$\mathrm{MAJ}_n^t(x_1,\ldots,x_n) = 1 \iff \sum_i x_i > t$$

(Monotone) Boolean Circuits

- Want to compute Boolean function $f\colon \{0,1\}^n \to \{0,1\}$
- Circuit is a directed acyclic graph
- Nodes are labeled with AND_2 and OR_2
- We are interested in the depth of the circuit
- The circuit below computes $\mathrm{MAJ}_3(x_1, x_2, x_3)$
- What is the minimal depth needed to compute MAJ_n?

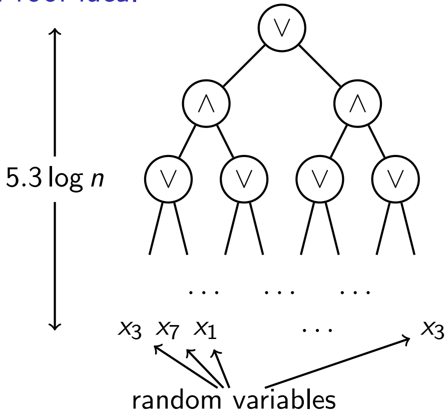


Valiant's Construction

Theorem (Valiant'84)

Majority can be computed by monotone formula of depth $5.3 \log n$

Proof idea.



□

Circuits for Majority

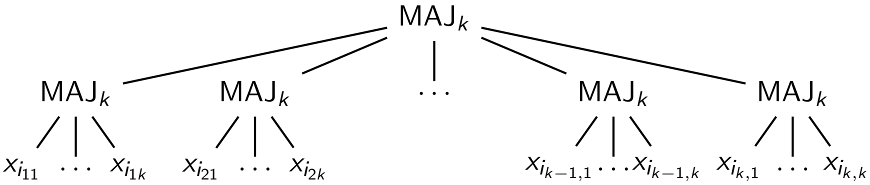
- Valiant's construction is simple, but is not explicit
- Another $O(\log n)$ upper bound by Ajtai, Komlós, Szemerédi, '83
  But there are drawbacks
  - Complicated construction
  - Large constant, impractical
- Still looking for a simple and practical construction
- Recent line of research: low-depth circuits for MAJ_n consisting of MAJ_k gates for $k < n$
- Motivation
  - Better understand the structure of majority
  - Possible iterative constructions

Majority Circuits

Problem

Compute MAJ_n *by circuits of constant depth* d *consisting of* MAJ_k*. What is the minimal* k *for which this is possible?*

Example for depth 2:



Simple Observations, $d = 2$

Problem

Compute MAJ_n by depth-2 circuits consisting of MAJ_k. What is the minimal k for which this is possible?

$1 \leqslant k \leqslant n$

Simple Observations, $d = 2$

Problem

Compute MAJ_n *by depth-2 circuits consisting of* MAJ_k*. What is the minimal* k *for which this is possible?*

$1 \leqslant k \leqslant n$

Observation

$k \geqslant n^{1/2}$

Simple Observations, $d = 2$

Problem

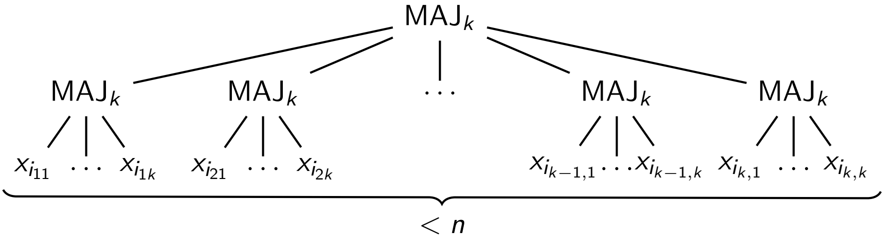
Compute MAJ_n *by depth-2 circuits consisting of* MAJ_k*. What is the minimal* k *for which this is possible?*

$1 \leqslant k \leqslant n$

Observation

$k \geqslant n^{1/2}$

Suppose $k < n^{1/2}$:



MAJ_n depends on all n variables, contradiction

Further Ideas, $d = 2$

Lemma

$k \geqslant cn^{2/3}$ for circuits with MAJ_k gates computing MAJ_n

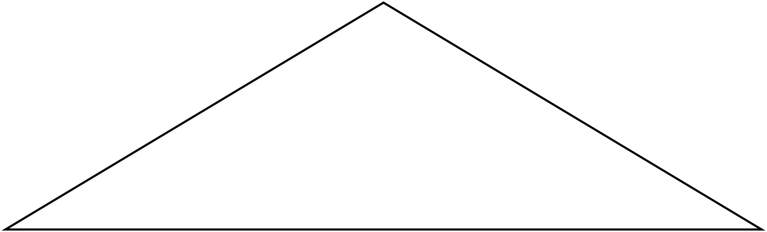
Further Ideas, $d = 2$

Lemma

$k \geqslant cn^{2/3}$ for circuits with MAJ_k gates computing MAJ_n

Proof.

Consider a circuit with $k \leqslant cn^{2/3}$ as a graph:



n variables, sum of degrees: $k^2 = n^{4/3}$ (ignore constants)

$\square$

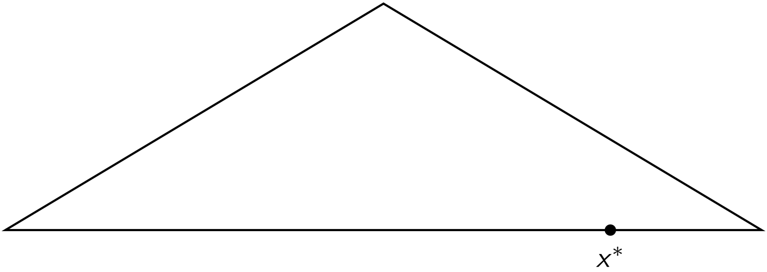
Further Ideas, $d = 2$

Lemma

$k \geqslant cn^{2/3}$ *for circuits with* MAJ_k *gates computing* MAJ_n

Proof.

Consider a circuit with $k \leqslant cn^{2/3}$ as a graph:



n variables, sum of degrees: $k^2 = n^{4/3}$ (ignore constants)

x^* — rare variable;

□

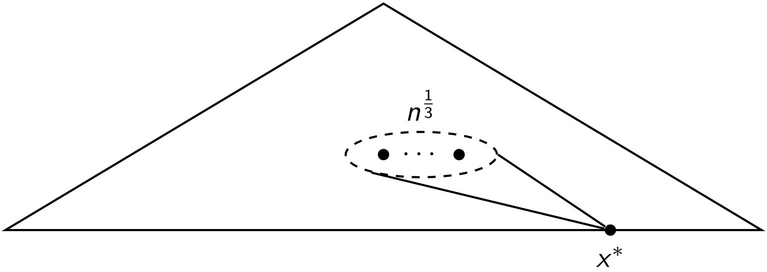
Further Ideas, $d = 2$

Lemma

$k \geqslant cn^{2/3}$ for circuits with MAJ_k gates computing MAJ_n

Proof.

Consider a circuit with $k \leqslant cn^{2/3}$ as a graph:



n variables, sum of degrees: $k^2 = n^{4/3}$ (ignore constants)
x^* — rare variable;

□

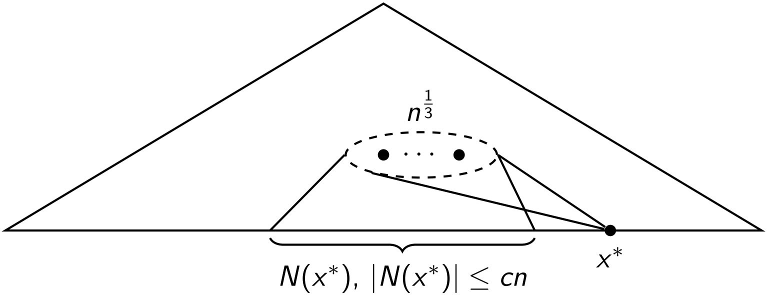
Further Ideas, $d = 2$

Lemma

$k \geqslant cn^{2/3}$ for circuits with MAJ_k gates computing MAJ_n

Proof.

Consider a circuit with $k \leqslant cn^{2/3}$ as a graph:



n variables, sum of degrees: $k^2 = n^{4/3}$ (ignore constants)
x^* — rare variable;

□

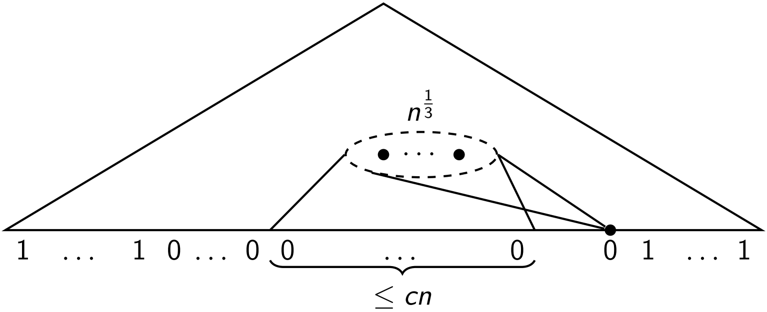
Further Ideas, $d = 2$

Lemma

$k \geqslant cn^{2/3}$ for circuits with MAJ_k gates computing MAJ_n

Proof.

Consider a circuit with $k \leqslant cn^{2/3}$ as a graph:



n variables, sum of degrees: $k^2 = n^{4/3}$ (ignore constants)

x^* — rare variable; fix $\vec{x}$ to a maxterm with x^* and $N(x^*)$ set to 0;

□

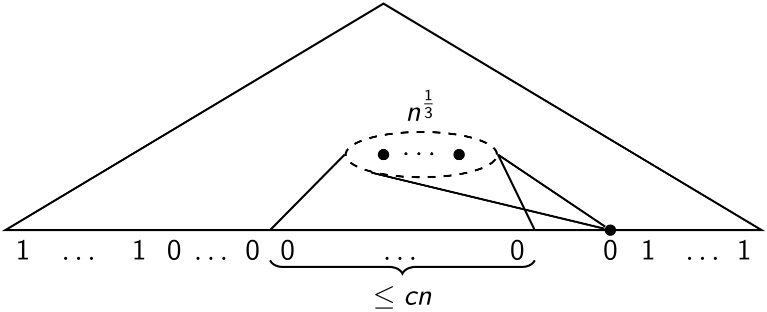
Further Ideas, $d = 2$

Lemma

$k \geqslant cn^{2/3}$ for circuits with MAJ_k gates computing MAJ_n

Proof.

Consider a circuit with $k \leqslant cn^{2/3}$ as a graph:



n variables, sum of degrees: $k^2 = n^{4/3}$ (ignore constants)
x^* — rare variable; fix $\vec{x}$ to a maxterm with x^* and $N(x^*)$ set to 0;
flip x^* to 1. Contradiction □

Results for Majority Circuits, $d = 2$

Theorem (Kulikov, P. '19)

$k = \widetilde{\Omega}(n^{2/3+1/57}) = \widetilde{\Omega}(n^{13/19})$ *for* $d = 2$

Theorem (Engels et al. '20)

$k \geqslant \Omega(n^{4/5})$ *for* $d = 2$ *with read-once gates on the bottom*

Theorem (Hrubes et al. '19)

$k \geqslant n/2 - o(n)$ *for* $d = 2$ *circuits with read-once gates on the bottom*

Theorem (Kombarov '18, Amano, Yoshida '18)

$k \leqslant n - 2$ *for* $d = 2$

Theorem (Posobin '17, Bauwens '17)

$k \leqslant 2n/3$ *for* $d = 2$ *circuits consisting of* MAJ_k^t

Results Majority Circuits, $d > 2$

Theorem (Kulikov, P. '19)
$k = O(n^{2/3})$ *for* $d = 3$

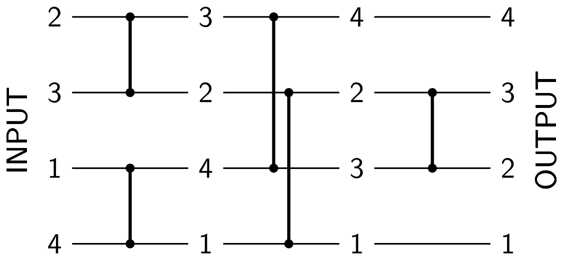
Theorem (Kulikov, P. '19)
$k = \widetilde{\Omega}\left(n^{26/(13d+12)}\right)$ *for* $d \geqslant 2$

A related direction for depth $d = 2$

Theorem (Lecomte et al. '22)
If $MAJ_n(x)$ *is computed as* $h(g_1(x), \ldots, g_t(x))$*, where* h *is an arbitrary function and* g_i*s are arbitrary functions depending on at most* k *variables, then* $t = \Omega(\frac{n}{k} \log k)$

Sorting Networks

- Input: $a = (a_1, \ldots, a_n) \in \mathbb{Z}^n$
- Output: Sorted a
- Elementary operations: comparators
  $(x, y) \mapsto (\min(x, y), \max(x, y))$
- Comparators are organized in layers
- Depth of the network: number of layers



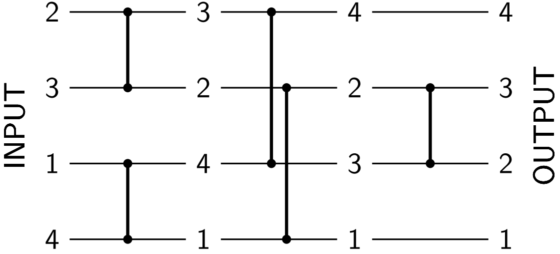
Sorting Networks, Motivation

- Sorting is a fundamental algorithmic task
- Sorting network is a simple sorting model
- Weaker than general comparison algorithms
- Can be implemented in hardware, convenient for parallelization
- Studied extensively since 1950s (a separate chapter in Knuth's book)
- Still there are important open problems

Zero-one Principle

Lemma

A network sorts all inputs $a \in \mathbb{Z}^n$ iff it sorts all inputs $a \in \{0, 1\}^n$



Connection to Majority Circuits

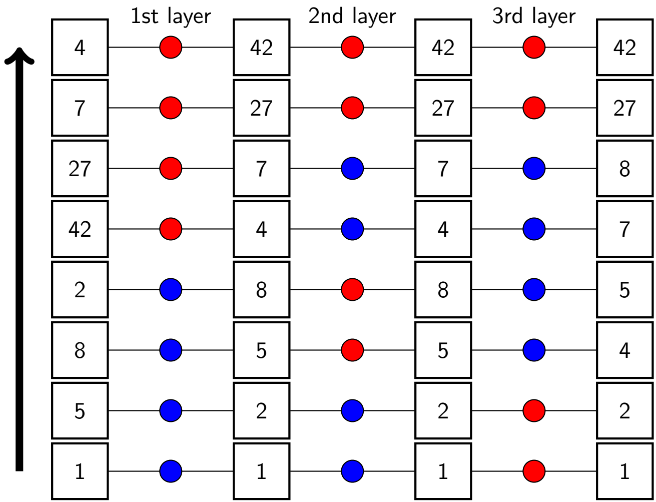
- Consider inputs in $\{0,1\}^n$
- Majority is just the middle bit of the sorted array
- Comparator computer AND and OR of its inputs: $\min(x,y) = \mathrm{AND}(x,y), \quad \max(x,y) = \mathrm{OR}(x,y)$
- Sorting network gives a monotone circuit for MAJ_n of the same depth

Known Results for Sorting Networks

- Many depth $O(\log^2 n)$ constructions (odd-even sorting network, pairwise sorting network, etc.)
- $O(\log n)$ upper bound (Ajtai, Komlós, Szemerédi, '83)
  But there are drawbacks
  - Complicated construction
  - Large constant, impractical
- Folklore lower bound: $\geqslant (2 - o(1)) \log n$
- Best known lower bound: $\approx 3.27 \log n$ (Kahale et al.'95)

k-Sorting Networks

- Generalization: comparators of arity $\leqslant k$, where k is a parameter



k-Sorting Networks, Motivation, Known Results

- Studied since 70s
- Motivation
  - Natural
  - Better understanding of sorting networks
  - Ideas for iterative constructions?
- Upper bound: $4 \log_k^2 n$ (Parker, Parbery, '89)
- Upper bound: $O(\log_k n)$ (Chvátal's lecture notes based on AKS construction)

Problem

Sort n inputs by k-sorters within constant depth d. What is the minimal k for which this is possible?

Results on Sorting Networks

Problem

Sort n inputs by k-sorters within constant depth d. What is the minimal k for which this is possible?

Lemma (Dobrokhotova-Maikova, Kozachinskiy, P. '22)

For $d \leqslant 2$ we have $k = n$

Theorem (Dobrokhotova-Maikova, Kozachinskiy, P. '22)

For $d = 3$ we have $k = \lceil \frac{n}{2} \rceil$

Theorem (Dobrokhotova-Maikova, Kozachinskiy, P. '22)

For $d = 4$ we have $k = \Theta(n^{2/3})$

Lemma (Dobrokhotova-Maikova, Kozachinskiy, P. '22)

$k \geqslant (\frac{n}{2})^{\frac{1}{\lceil d/2 \rceil}}$ *for arbitrary d*

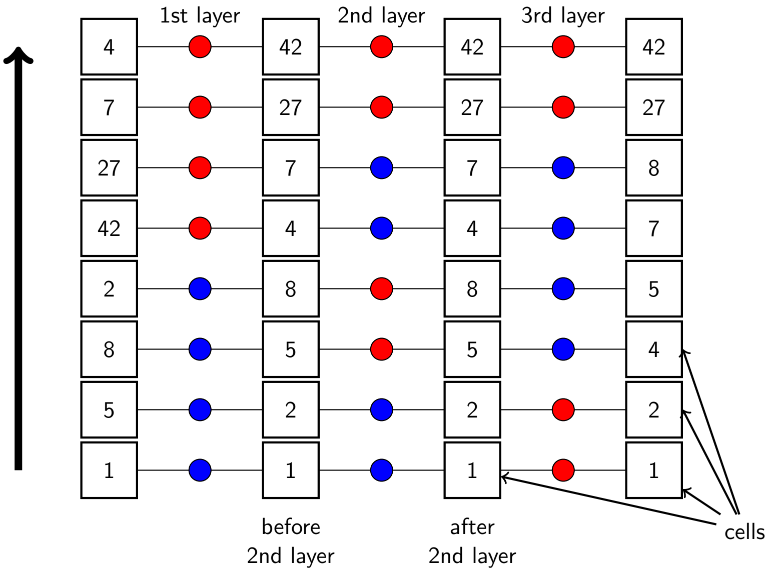
Comparison

Depth	Majority	Sorting
$d = 2$	$\frac{n}{2} \leqslant k \leqslant \frac{2n}{3}$	$k = n$
$d = 3$	$O(n^{2/3})$	$k = \lceil \frac{n}{2} \rceil$
$d = 4$	$k = \widetilde{\Omega}\left(n^{13/32}\right)$	$\Theta(n^{2/3})$
$d \geqslant 5$	$k = \widetilde{\Omega}\left(n^{26/(13d+12)}\right)$	$k \geqslant \left(\frac{n}{2}\right)^{\frac{1}{\lceil d/2 \rceil}}$

Comparison

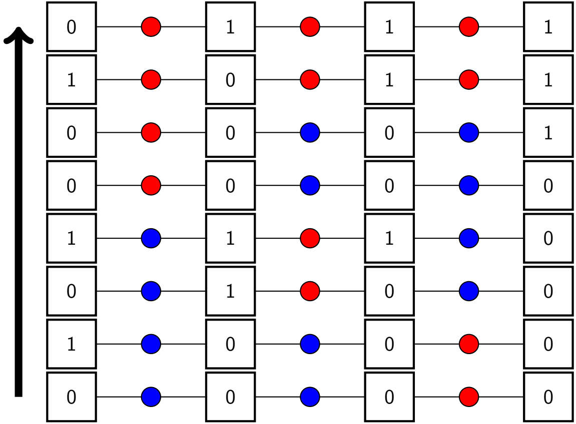
Depth	Read-once Majority	Sorting
$d = 2$	$\frac{n}{2} \leqslant k \leqslant \frac{2n}{3}$	$k = n$
$d = 3$	$O(n^{2/3})$	$k = \lceil \frac{n}{2} \rceil$
$d = 4$	$k = \widetilde{\Omega}\left(n^{7/17}\right)$	$\Theta(n^{2/3})$
$d \geqslant 5$	$k = \widetilde{\Omega}\left(n^{14/(7d+6)}\right)$	$k \geqslant \left(\frac{n}{2}\right)^{\frac{1}{\lceil d/2 \rceil}}$

Notation


1st layer
2nd layer
3rd layer
4
7
27
42
2
8
5
1
42
27
7
4
8
5
2
1
42
27
7
4
8
5
2
1
42
27
8
7
5
4
2
1
before
2nd layer
after
2nd layer
cells

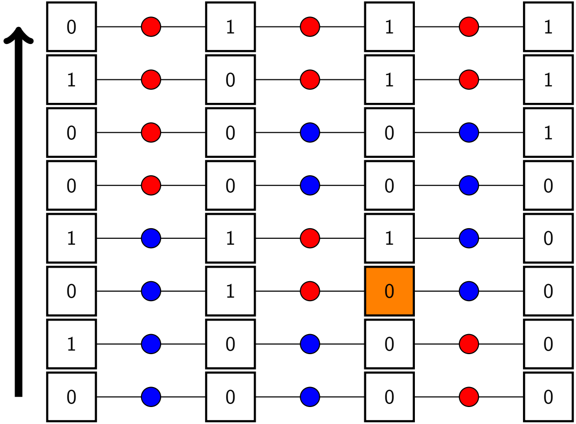
Proof Idea: Access

- Consider a sorting network, fix input $x \in \{0,1\}^n$
- We have *access* to some cell if
  - It contains 0
  - We can switch some x_i from 0 to 1 and the cell will switch to 1



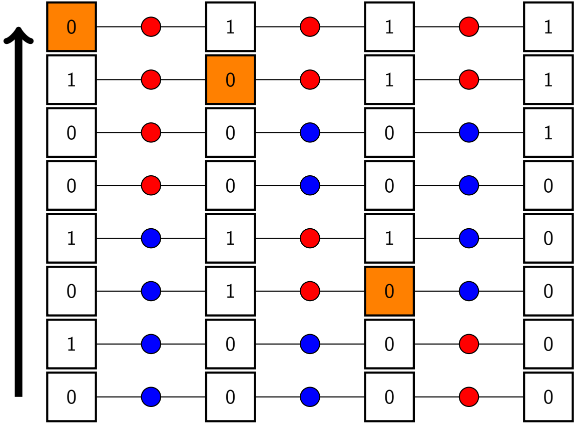
Proof Idea: Access

- Consider a sorting network, fix input $x \in \{0, 1\}^n$
- We have *access* to some cell if
  - It contains 0
  - We can switch some x_i from 0 to 1 and the cell will switch to 1



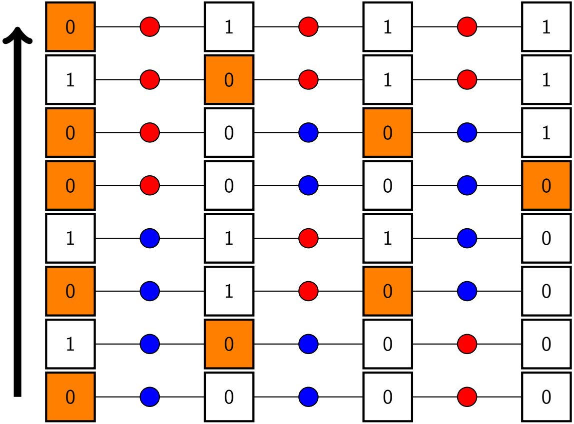
Proof Idea: Access

- Consider a sorting network, fix input $x \in \{0,1\}^n$
- We have *access* to some cell if
  - It contains 0
  - We can switch some x_i from 0 to 1 and the cell will switch to 1



Proof Idea: Access

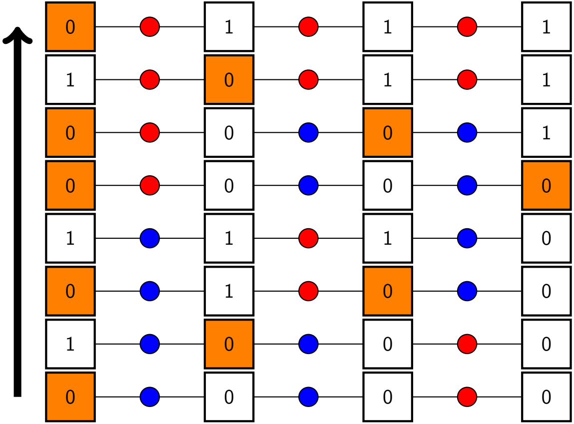
- Consider a sorting network, fix input $x \in \{0, 1\}^n$
- We have *access* to some cell if
  - It contains 0
  - We can switch some x_i from 0 to 1 and the cell will switch to 1



Proof Idea: Access

Lemma (Access stability)

Suppose we have access to some cell c and we switch some x_i from 0 to 1. If c still contains 0, we still have access to it



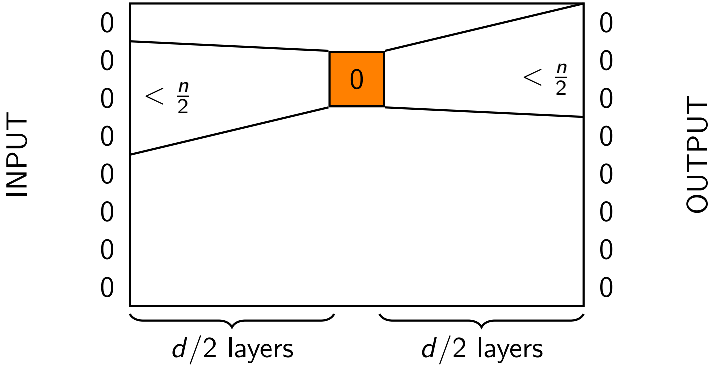
Lower Bound for Arbitrary d

Lemma

For any sorting network with parameters n, k and d we have $k \geqslant \left(\frac{n}{2}\right)^{\frac{1}{\lceil d/2 \rceil}}$.

Proof idea.

Assume d is even and $k^{d/2} < \frac{n}{2}$



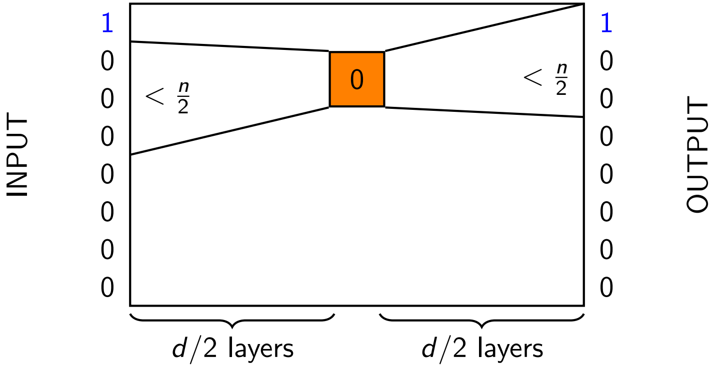
Lower Bound for Arbitrary d

Lemma

For any sorting network with parameters n, k and d we have $k \geqslant \left(\frac{n}{2}\right)^{\frac{1}{\lceil d/2 \rceil}}$.

Proof idea.

Assume d is even and $k^{d/2} < \frac{n}{2}$



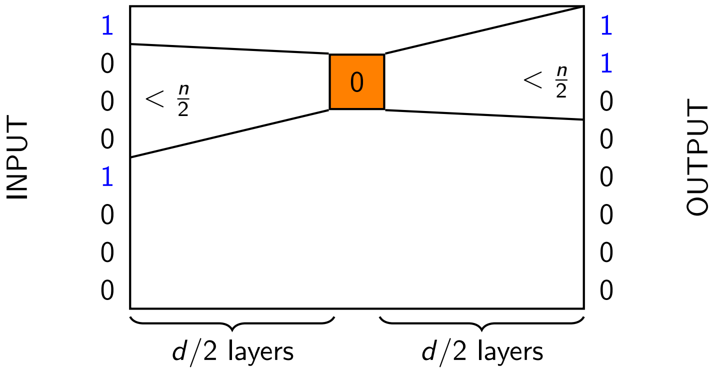
Lower Bound for Arbitrary d

Lemma

For any sorting network with parameters n, k and d we have $k \geqslant \left(\frac{n}{2}\right)^{\frac{1}{\lceil d/2 \rceil}}$.

Proof idea.

Assume d is even and $k^{d/2} < \frac{n}{2}$



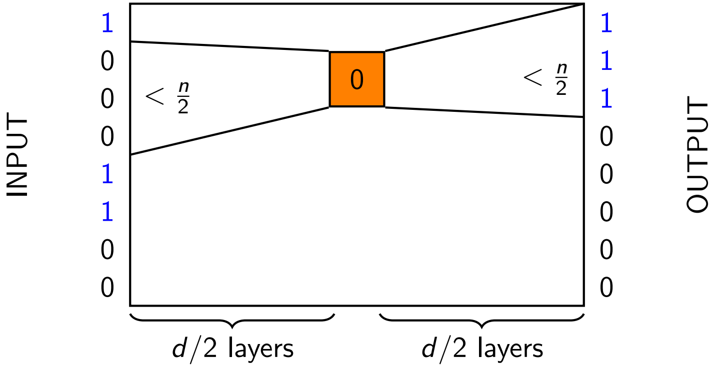
Lower Bound for Arbitrary d

Lemma

For any sorting network with parameters n, k and d we have
$k \geqslant \left(\frac{n}{2}\right)^{\frac{1}{\lceil d/2 \rceil}}$.

Proof idea.

Assume d is even and $k^{d/2} < \frac{n}{2}$



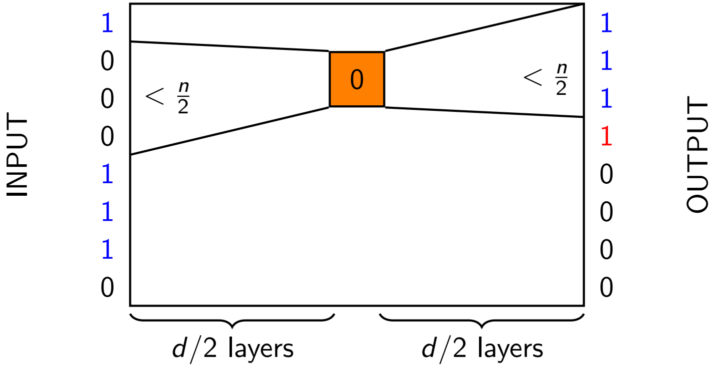
Lower Bound for Arbitrary d

Lemma

For any sorting network with parameters n, k and d we have $k \geqslant \left(\frac{n}{2}\right)^{\frac{1}{\lceil d/2 \rceil}}$.

Proof idea.

Assume d is even and $k^{d/2} < \frac{n}{2}$



Constant d, Initial Ideas

Consider a network with parameters n, k and d. How can we tell that it is incorrect?

Lemma

If on some input $x \in \{0,1\}^n$ we have access to two cells after the last layer, the network is incorrect

Corollary

For $d = 1$ we have $k = n$

Constant d, Initial Ideas

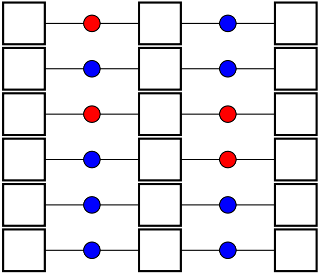
Is there a condition in terms of previous levels of cells?

Lemma

If on some input $x \in \{0,1\}^n$ we have access to two cells before the last layer that go to different comparators, the network is incorrect

Corollary

For $d = 2$ we have $k = n$



Puzzle: in the store there are vases of different colors and shapes. Show that there are two vases that differ both in color and shape

Proof Idea: Growing Branch

For larger depth we need a condition that does not address the last layer of comparators

Definition (Growing Branch)

A sequence of inputs $x^1, \ldots, x^p \in \{0,1\}^n$ is a *growing branch* if for each i x^{i+1} is obtained from x^i by changing one coordinate from 0 to 1

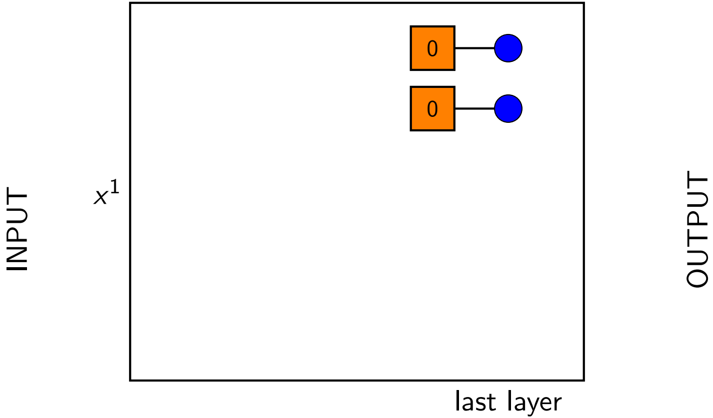
Lemma

Consider a sorting network with parameters n, k, d. Assume that there is a growing branch $x^1, \ldots, x^k$ such that for every x^i we have access to at least two cells before the last layer. Then the network is incorrect

Proof Idea: Growing Branch

Lemma (Restated)

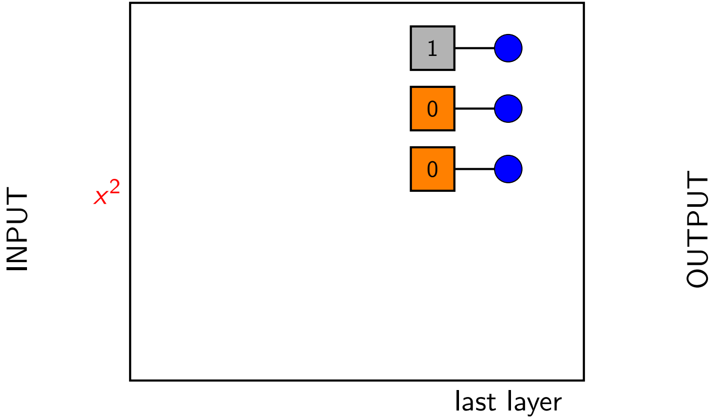
Consider a sorting network with parameters n, k, d. Assume that there is a growing branch $x^1, \ldots, x^k$ such that for every x^i we have access to at least two cells before the last layer. Then the network is incorrect



Proof Idea: Growing Branch

Lemma (Restated)

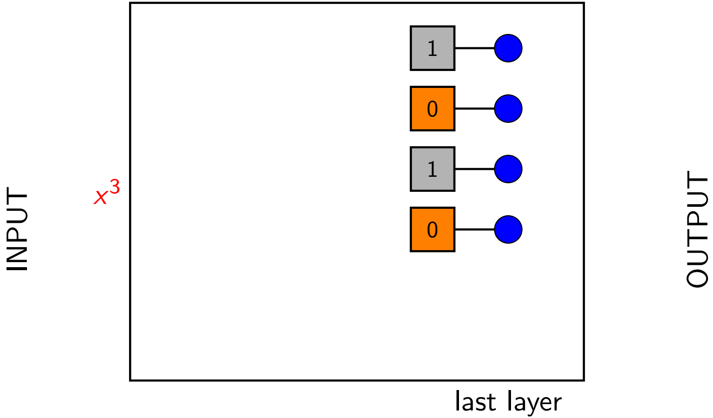
Consider a sorting network with parameters n, k, d. Assume that there is a growing branch $x^1, \ldots, x^k$ such that for every x^i we have access to at least two cells before the last layer. Then the network is incorrect



Proof Idea: Growing Branch

Lemma (Restated)

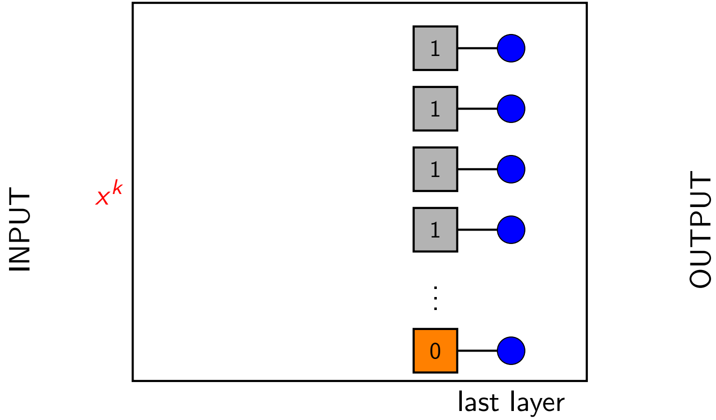
Consider a sorting network with parameters n, k, d. Assume that there is a growing branch $x^1, \ldots, x^k$ such that for every x^i we have access to at least two cells before the last layer. Then the network is incorrect



Proof Idea: Growing Branch

Lemma (Restated)

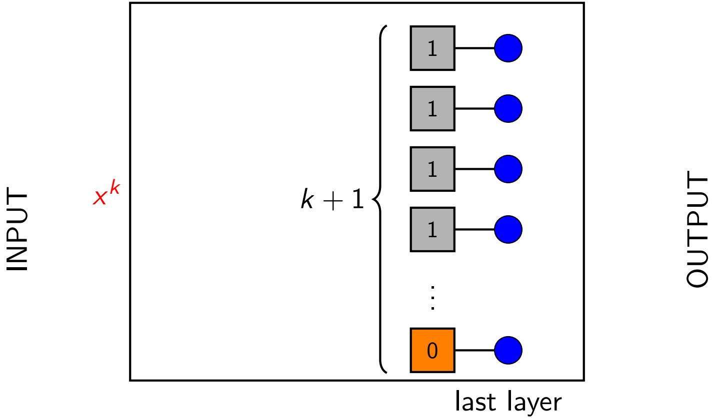
Consider a sorting network with parameters n, k, d. Assume that there is a growing branch $x^1, \ldots, x^k$ such that for every x^i we have access to at least two cells before the last layer. Then the network is incorrect



Proof Idea: Growing Branch

Lemma (Restated)

Consider a sorting network with parameters n, k, d. Assume that there is a growing branch $x^1, \ldots, x^k$ *such that for every* x^i *we have access to at least two cells before the last layer. Then the network is incorrect*



Proof Idea: Growing Branch

Lemma (Restated)

Consider a sorting network with parameters n, k, d. Assume that there is a growing branch $x^1, \ldots, x^k$ such that for every x^i we have access to at least two cells before the d-th layer. Then the network is incorrect

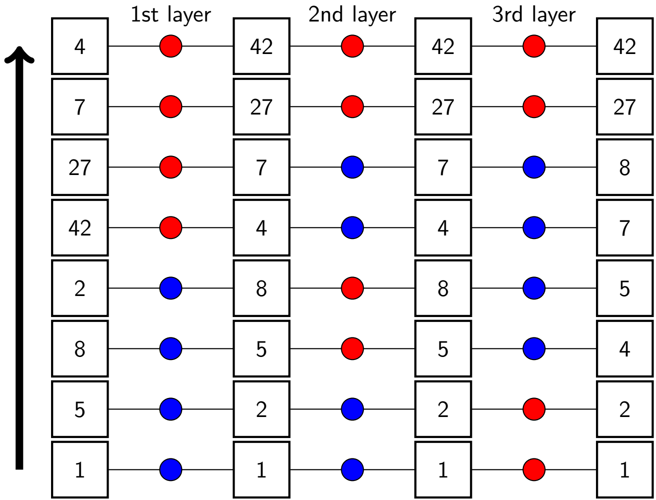
Corollary

Consider a sorting network with parameters n, k, d. Assume that there is a growing branch $x^1, \ldots, x^k$ such that for every x^i we have access to at least two cells before the $(d-1)$-th layer going to different comparators. Then the network is incorrect

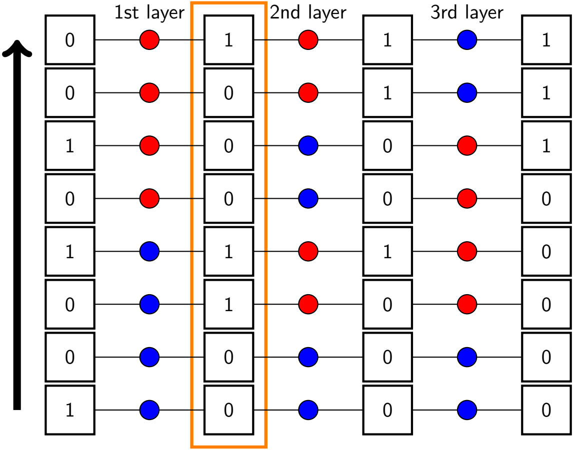
Depth $d = 3$, Upper Bound

Lemma

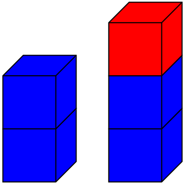
For $d = 3$ we have $k \leqslant \lceil \frac{n}{2} \rceil$



Depth $d = 3$, Lower Bound, Cube Puzzle



Cube stacks — 1st layer comparators
Cube colors — 2nd layer comparators



Cube stacks

Depth $d = 3$, Lower Bound, Cube Puzzle

The puzzle:

- n cubes are arranged in vertical stacks
- Each stack is of size at most k
- Each cube has a color
- For each color there are at most k cubes of this color
- In one step we can take one cube from the top of one of the stacks
- Show that we can have cubes of different colors on the tops for at least k steps in a row (we can take some cubes before we get this property)

Relevance:

- This is exactly a growing branch of length k
- Implies that there is no sorting network with parameters n, k and $d = 3$

Cube Puzzle

Theorem

Cube puzzle is solvable for $k < \lceil \frac{n}{2} \rceil$

Corollary

For the sorting networks of depth $d = 3$ *we have* $k \geqslant \lceil \frac{n}{2} \rceil$

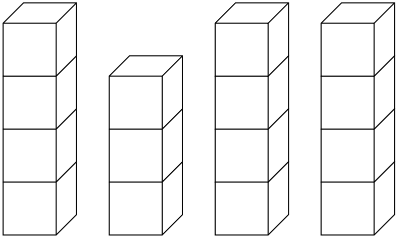
Lemma (Weaker form)

Cube puzzle is solvable for $k \leqslant \frac{n}{3}$

Cube Puzzle

Lemma (Weaker form, restated)

Cube puzzle is solvable for $k \leqslant \frac{n}{3}$

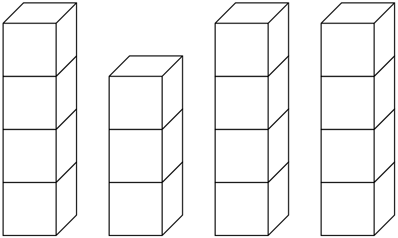


Cube Puzzle

Lemma (Weaker form, restated)

Cube puzzle is solvable for $k \leqslant \frac{n}{3}$

- Find cubes of different colors in different stacks

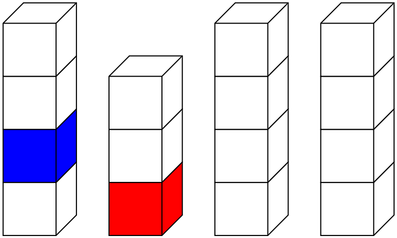


Cube Puzzle

Lemma (Weaker form, restated)

Cube puzzle is solvable for $k \leqslant \frac{n}{3}$

- Find cubes of different colors in different stacks

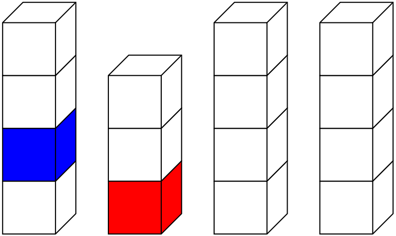


Cube Puzzle

Lemma (Weaker form, restated)

Cube puzzle is solvable for $k \leqslant \frac{n}{3}$

- Find cubes of different colors in different stacks
- Take all cubes above them

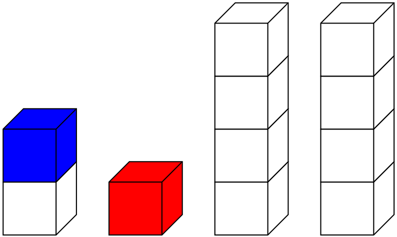


Cube Puzzle

Lemma (Weaker form, restated)

Cube puzzle is solvable for $k \leqslant \frac{n}{3}$

- Find cubes of different colors in different stacks
- Take all cubes above them

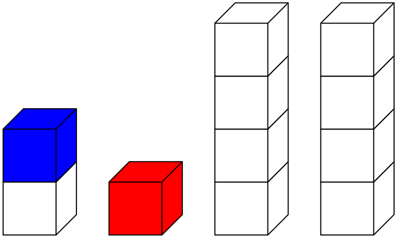


Cube Puzzle

Lemma (Weaker form, restated)

Cube puzzle is solvable for $k \leqslant \frac{n}{3}$

- Find cubes of different colors in different stacks
- Take all cubes above them
- Now we have two tops of different colors

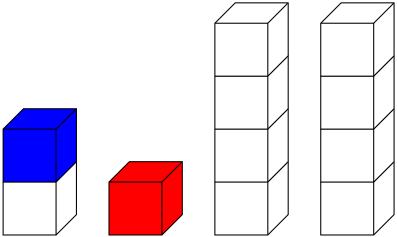


Cube Puzzle

Lemma (Weaker form, restated)

Cube puzzle is solvable for $k \leqslant \frac{n}{3}$

- Find cubes of different colors in different stacks
- Take all cubes above them
- Now we have two tops of different colors
- Take all cubes from other stacks one by one

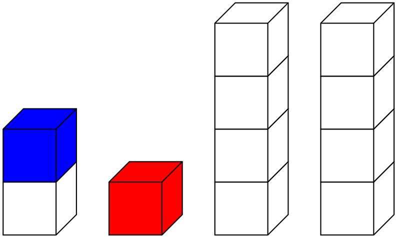


Cube Puzzle

Lemma (Weaker form, restated)

Cube puzzle is solvable for $k \leqslant \frac{n}{3}$

- Find cubes of different colors in different stacks
- Take all cubes above them
- Now we have two tops of different colors
- Take all cubes from other stacks one by one
- There are $\geqslant \frac{n}{3} \geqslant k$ of them!



Depth $d = 4$

Theorem (Implicit in Leighton '85 (ColumnSort))

There is a sorting network of depth $d = 4$ with $k = O(n^{2/3})$

Theorem

For depth $d = 4$ we have $k = \Omega(n^{2/3})$

Proof idea.

A more complicated version of a cube puzzle
Probabilistic proof of solvability □

Conclusion

Problem

Compute MAJ_n by circuits of constant depth d consisting of MAJ_k. What is the minimal k for which this is possible?

Problem

Sort n inputs by k-sorters within constant depth d. What is the minimal k for which this is possible?

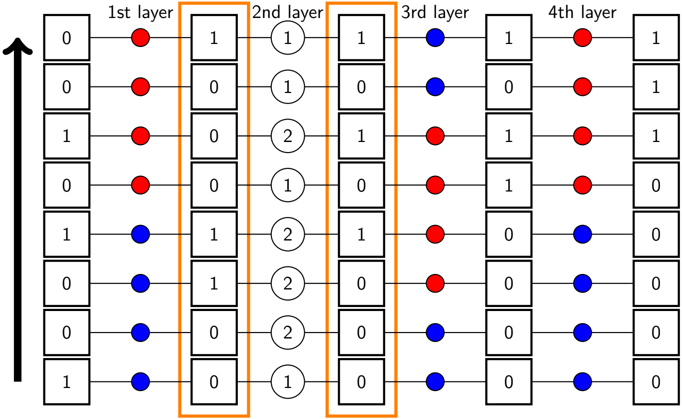
Depth	Majority	Sorting
$d = 2$	$\frac{n}{2} \leqslant k \leqslant \frac{2n}{3}$	$k = n$
$d = 3$	$O(n^{2/3})$	$k = \lceil \frac{n}{2} \rceil$
$d = 4$	$k = \widetilde{\Omega}\left(n^{13/32}\right)$	$\Theta(n^{2/3})$
$d \geqslant 5$	$k = \widetilde{\Omega}\left(n^{26/(13d+12)}\right)$	$k \geqslant \left(\frac{n}{2}\right)^{\frac{1}{\lceil d/2 \rceil}}$

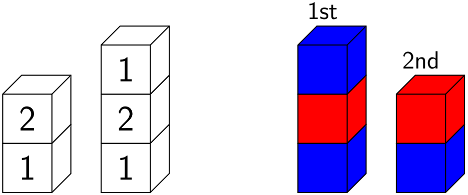
Conclusion

Depth	Majority	Sorting
$d = 2$	$\frac{n}{2} \leqslant k \leqslant \frac{2n}{3}$	$k = n$
$d = 3$	$O(n^{2/3})$	$k = \lceil \frac{n}{2} \rceil$
$d = 4$	$k = \widetilde{\Omega}\left(n^{13/32}\right)$	$\Theta(n^{2/3})$
$d \geqslant 5$	$k = \widetilde{\Omega}\left(n^{26/(13d+12)}\right)$	$k \geqslant \left(\frac{n}{2}\right)^{\frac{1}{\lceil d/2 \rceil}}$

Thank you!

Depth $d = 4$, Cube Puzzle-2





Depth $d = 4$, Cube Puzzle-2

- There are two sets of cubes, n left cubes and n right cubes
- Both sets are arranged in stacks of size at most k
- Left cubes are labeled with numbers, right cubes are colored
- For each color or number there are at most k cubes of this color or number
- The numbers on left cubes are in one to one correspondence with stacks of the right cubes
- In one step we can remove a top cube from one left stack. If its label is i, we also remove the top cube from ith stack on the right
- A top left cube with label i gives access to the color of the top cube in the ith stack on the right
- Show that we can have access to at least two different colors at least k steps in a row

Depth $d = 4$, Cube Puzzle-2

Theorem

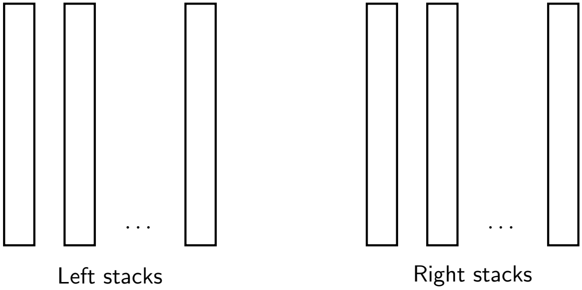
Cube puzzle-2 is solvable for $k = O(n^{2/3})$

Corollary

For sorting networks of depth $d = 4$ *we have* $k = \Omega(n^{2/3})$

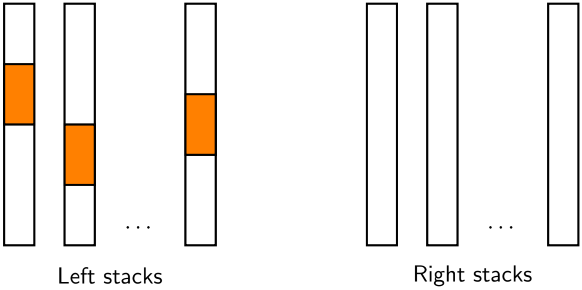
Cube Puzzle-2, Proof Idea

- We have $\frac{n}{k} = 100n^{1/3}$ stacks of size $k = \frac{n^{2/3}}{100}$ on each side



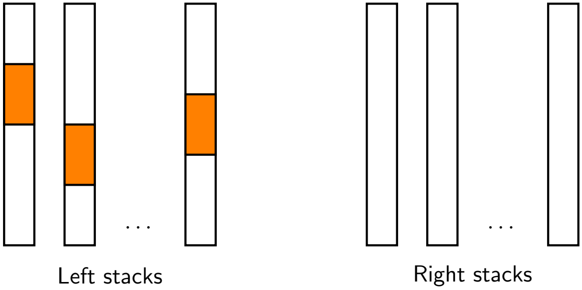
Cube Puzzle-2, Proof Idea

- We have $\frac{n}{k} = 100n^{1/3}$ stacks of size $k = \frac{n^{2/3}}{100}$ on each side
- Let $l = n^{1/3}$ and pick random length l interval on the left



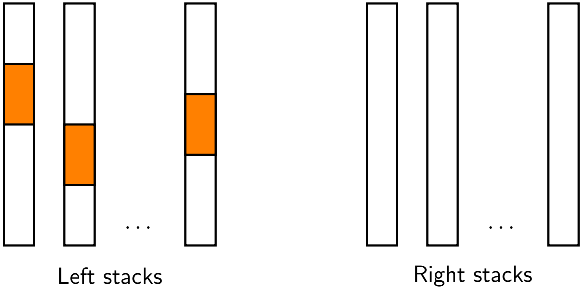
Cube Puzzle-2, Proof Idea

- We have $\frac{n}{k} = 100n^{1/3}$ stacks of size $k = \frac{n^{2/3}}{100}$ on each side
- Let $l = n^{1/3}$ and pick random length l interval on the left
- With high probability for each i only a small fraction of intervals contain cubes labeled with i



Cube Puzzle-2, Proof Idea

- We have $\frac{n}{k} = 100n^{1/3}$ stacks of size $k = \frac{n^{2/3}}{100}$ on each side
- Let $l = n^{1/3}$ and pick random length l interval on the left
- With high probability for each i only a small fraction of intervals contain cubes labeled with i
- Take all cubes above the intervals



Cube Puzzle-2, Proof Idea

- We have $\frac{n}{k} = 100n^{1/3}$ stacks of size $k = \frac{n^{2/3}}{100}$ on each side
- Let $l = n^{1/3}$ and pick random length l interval on the left
- With high probability for each i only a small fraction of intervals contain cubes labeled with i
- Take all cubes above the intervals
- If we have access to some color in the i-th stack on the right and we want to keep it, just do not touch intervals that contain labels i