MATRIX RIGIDITY

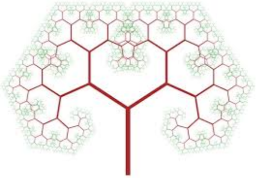
RIGIDITY AND DATA STRUCTURES

Sasha Golovnev

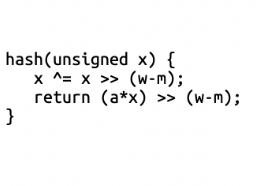
December 3, 2020

EXAMPLES

Stack, Queue, List, Heap



Search Trees



```
hash(unsigned x) {
    x ^= x >> (w-m);
    return (a*x) >> (w-m);
}
```

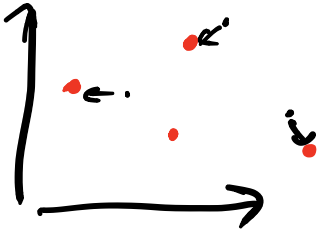
Hash Tables

STATIC DATA STRUCTURES. EXAMPLES

- Graph Distances: Preprocess a road network in order to efficiently compute distances between cities
  (Google Maps)

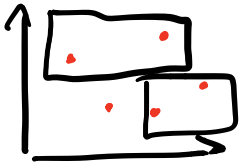
STATIC DATA STRUCTURES. EXAMPLES

- Graph Distances: Preprocess a road network in order to efficiently compute distances between cities
(Google Maps)
- Nearest Neighbors: Preprocess a set of points in order to efficiently find closest point to a query point
(Netflix recommendations)



STATIC DATA STRUCTURES. EXAMPLES

- Graph Distances: Preprocess a road network in order to efficiently compute distances between cities
  (Google Maps)
- Nearest Neighbors: Preprocess a set of points in order to efficiently find closest point to a query point
  (Netflix recommendations)
- Range Counting: Preprocess a set of points in order to efficiently compute the number of points in a given rectangle
  (Amazon market size estimation)



STATIC DATA STRUCTURES

1	0	1	0	1	1	1	1	0	0	0	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	1	1
0	0	1	1	1	0	0	1	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	1	0	1	0	1	0	1	1	0	0	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1	0	1	1
0	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0	0	0	1	0	1	0	1	1	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	1
1	1	1	0	0	1	0	0	0	1	0	0	1	1	0	1	0	1	0	0	1	0	1	1	1	1	1	0	1	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	1	0	1	1	1	0	1
0	1	0	1	1	1	0	1	1	1	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	0	1	1	0	0	0	1	1	1	1	0	0	1	0	1	1	1	0	0	0	1	0	1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	1	0	1	1	0
1	0	1	1	0	1	1	0	1	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	1	1	1	0	1	0	0	0	0	1	0	0	1	0	1	1	0	1	0	1	1	0	0	0	1	0	1	1	1
0	1	1	0	1	0	1	1	0	1	1	0	1	1	1	1	0	1	0	0	1	0	0	1	0	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0
0	1	1	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	1	0	0	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	0	0	0	1	0	1	0	0	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1	0	1	0
1	1	0	0	1	0	1	0	0	0	1	1	1	0	1	0	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1	1	0	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1	0	0	0	0
1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	1	0	1	1	0	0	0	1	1	1	0	1	0	0	1	0	1	1	1	0	1	0	1	0

unbounded

Preprocessing

STATIC DATA STRUCTURES

Queries

1	0	1	0	1	1	1	1	0	0	0	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	1	1
0	0	1	1	1	0	0	1	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	1	0	1	0	1	0	1	1	0	0	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1	0	1	1
0	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0	0	0	1	0	1	0	1	1	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	1
1	1	1	0	0	1	0	0	0	1	0	0	1	1	0	1	0	1	0	0	1	0	1	1	1	1	1	0	1	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	1	0	1	1	1	0	1
0	1	0	1	1	1	0	1	1	1	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	0	1	1	0	0	0	1	1	1	1	0	0	1	0	1	1	1	0	0	0	1	0	1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	1	0	1	1	0
1	0	1	1	0	1	1	0	1	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	1	1	1	0	1	0	0	0	0	1	0	0	1	0	1	1	0	1	0	1	1	0	0	0	1	0	1	1	1
0	1	1	0	1	0	1	1	0	1	1	0	1	1	1	1	0	1	0	0	1	0	0	1	0	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0
0	1	1	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	1	0	0	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	0	0	0	1	0	1	0	0	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1	0	1	0
1	1	0	0	1	0	1	0	0	0	1	1	1	0	1	0	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1	1	0	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1	0	0	0	0
1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	1	0	1	1	0	0	0	1	1	1	0	1	0	0	1	0	1	1	1	0	1	0	1	0

Preprocessing

STATIC DATA STRUCTURES

Queries

Boston — Washington

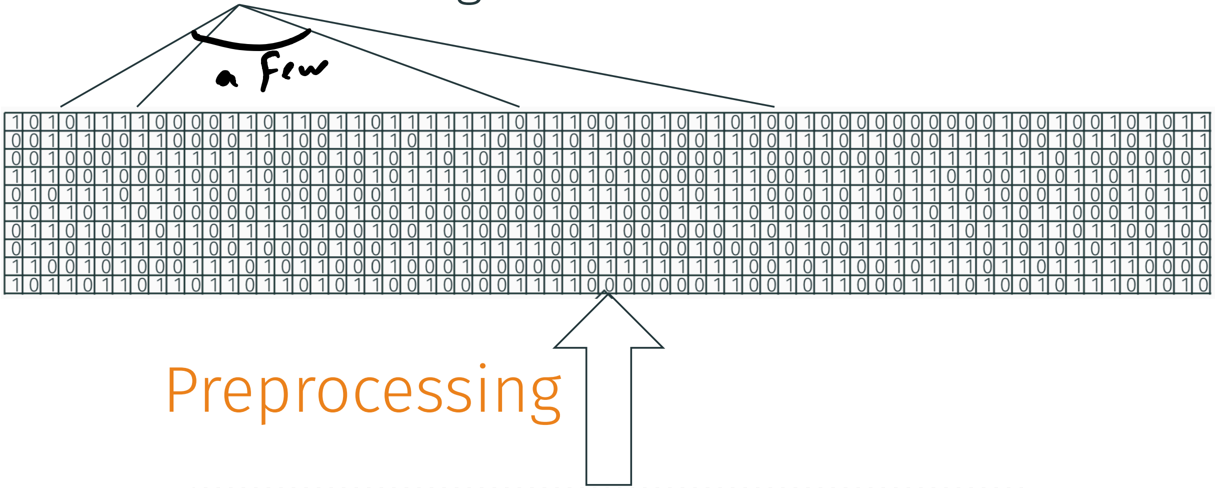
1	0	1	0	1	1	1	1	0	0	0	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	1	1
0	0	1	1	1	0	0	1	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	1	0	1	0	1	0	1	1	0	0	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1	0	1	1
0	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0	0	0	1	0	1	0	1	1	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	1
1	1	1	0	0	1	0	0	0	1	0	0	1	1	0	1	0	1	0	0	1	0	1	1	1	1	1	0	1	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	1	0	1	1	1	0	1
0	1	0	1	1	1	0	1	1	1	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	0	1	1	0	0	0	1	1	1	1	0	0	1	0	1	1	1	0	0	0	1	0	1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	1	0	1	1	0
1	0	1	1	0	1	1	0	1	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	1	1	1	0	1	0	0	0	0	1	0	0	1	0	1	1	0	1	0	1	1	0	0	0	1	0	1	1	1
0	1	1	0	1	0	1	1	0	1	1	0	1	1	1	1	0	1	0	0	1	0	0	1	0	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0
0	1	1	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	1	0	0	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	0	0	0	1	0	1	0	0	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1	0	1	0
1	1	0	0	1	0	1	0	0	0	1	1	1	0	1	0	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1	1	0	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1	0	0	0	0
1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	1	0	1	1	0	0	0	1	1	1	0	1	0	0	1	0	1	1	1	0	1	0	1	0

Preprocessing

STATIC DATA STRUCTURES

Queries

Boston — Washington



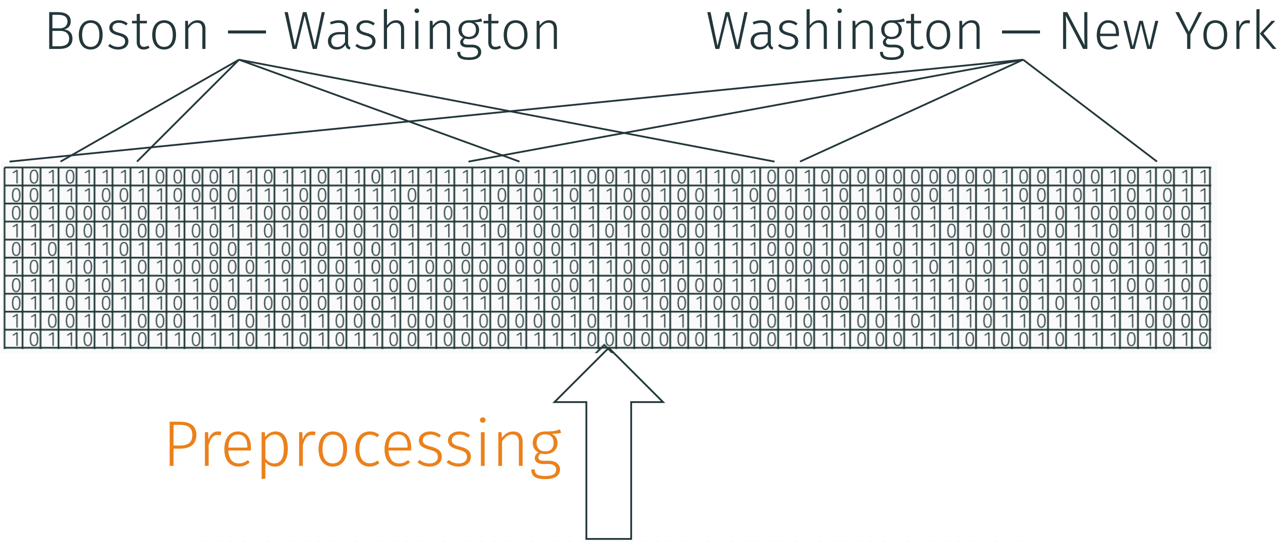
a few

1	0	1	0	1	1	1	1	0	0	0	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	1	1
0	0	1	1	1	0	0	1	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	1	0	1	0	1	0	1	1	0	0	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1	0	1	1
0	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0	0	0	1	0	1	0	1	1	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	1
1	1	1	0	0	1	0	0	0	1	0	0	1	1	0	1	0	1	0	0	1	0	1	1	1	1	1	0	1	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	1	0	1	1	1	0	1
0	1	0	1	1	1	0	1	1	1	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	0	1	1	0	0	0	1	1	1	1	0	0	1	0	1	1	1	0	0	0	1	0	1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	1	0	1	1	0
1	0	1	1	0	1	1	0	1	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	1	1	1	0	1	0	0	0	0	1	0	0	1	0	1	1	0	1	0	1	1	0	0	0	1	0	1	1	1
0	1	1	0	1	0	1	1	0	1	1	0	1	1	1	1	0	1	0	0	1	0	0	1	0	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0
0	1	1	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	1	0	0	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	0	0	0	1	0	1	0	0	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1	0	1	0
1	1	0	0	1	0	1	0	0	0	1	1	1	0	1	0	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1	1	0	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1	0	0	0	0
1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	1	0	1	1	0	0	0	1	1	1	0	1	0	0	1	0	1	1	1	0	1	0	1	0

Preprocessing

STATIC DATA STRUCTURES

Queries



Boston — Washington

Washington — New York

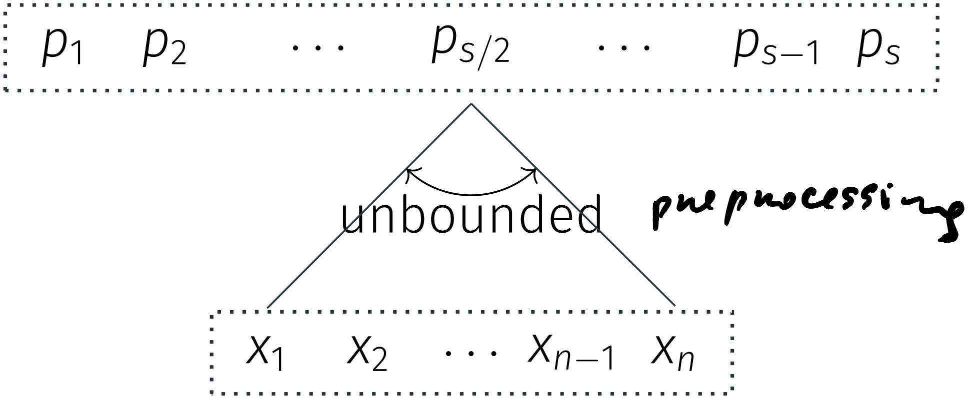
1	0	1	0	1	1	1	1	0	0	0	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	1	1
0	0	1	1	1	0	0	1	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	1	0	1	0	1	0	1	1	0	0	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1	0	1	1
0	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0	0	0	1	0	1	0	1	1	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	1
1	1	1	0	0	1	0	0	0	1	0	0	1	1	0	1	0	1	0	0	1	0	1	1	1	1	1	0	1	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	1	0	1	1	1	0	1
0	1	0	1	1	1	0	1	1	1	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	0	1	1	0	0	0	1	1	1	1	0	0	1	0	1	1	1	0	0	0	1	0	1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	1	0	1	1	0
1	0	1	1	0	1	1	0	1	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	1	1	1	0	1	0	0	0	0	1	0	0	1	0	1	1	0	1	0	1	1	0	0	0	1	0	1	1	1
0	1	1	0	1	0	1	1	0	1	1	0	1	1	1	1	0	1	0	0	1	0	0	1	0	1	0	0	1	1	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0
0	1	1	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	1	0	0	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	0	0	0	1	0	1	0	0	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1	0	1	0
1	1	0	0	1	0	1	0	0	0	1	1	1	0	1	0	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1	1	0	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1	0	0	0	0
1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	1	0	1	1	0	0	0	1	1	1	0	1	0	0	1	0	1	1	1	0	1	0	1	0

Preprocessing

STATIC DATA STRUCTURES. DEFINITION

input $x_1 \quad x_2 \quad \cdots \quad x_{n-1} \quad x_n$ — Road networks

STATIC DATA STRUCTURES. DEFINITION



p_1 p_2 $\cdots$ $p_{s/2}$ $\cdots$ p_{s-1} p_s

s memory cells

unbounded

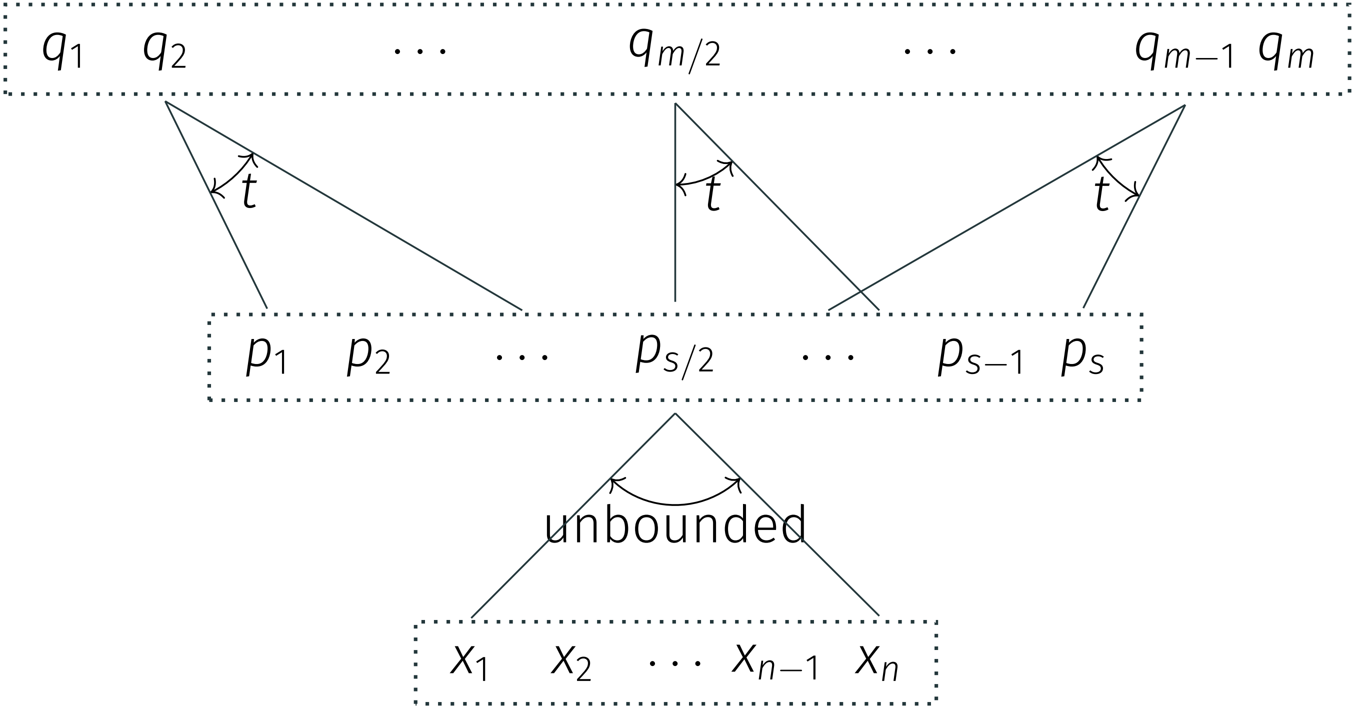
preprocessing

x_1 x_2 $\cdots$ x_{n-1} x_n

STATIC DATA STRUCTURES. DEFINITION

m queries are known in advance

$m = n^{O(1)}$ Say, $m = n^{100}$



q_1 q_2 $\cdots$ $q_{m/2}$ $\cdots$ q_{m-1} q_m

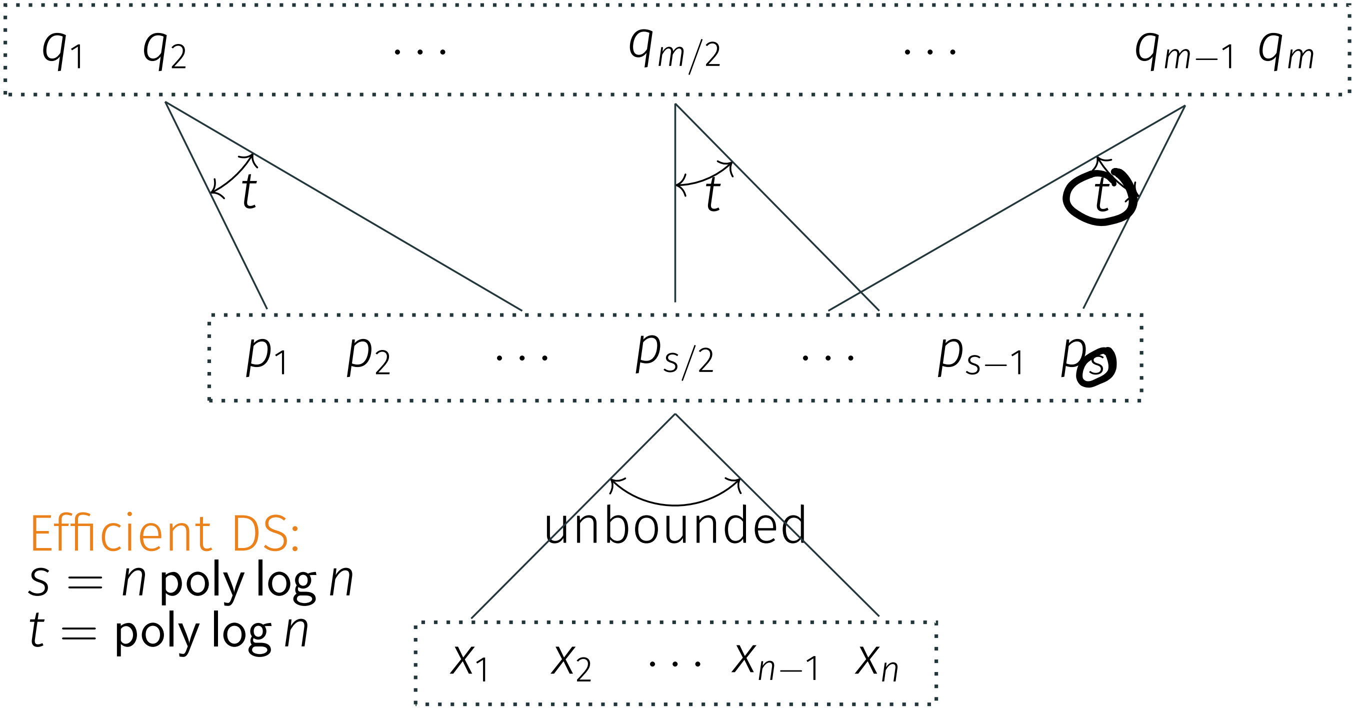
t t t

p_1 p_2 $\cdots$ $p_{s/2}$ $\cdots$ p_{s-1} p_s

unbounded

x_1 x_2 $\cdots$ x_{n-1} x_n

STATIC DATA STRUCTURES. DEFINITION



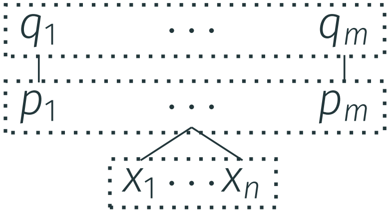
Efficient DS:
$s = n \operatorname{poly}\log n$
$t = \operatorname{poly}\log n$

DS LOWER BOUNDS

- Two trivial solutions:

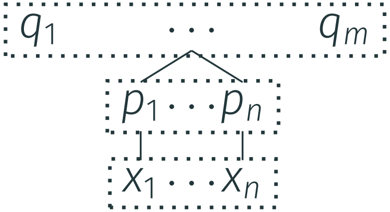
DS Lower Bounds

- Two trivial solutions:
  - $s = \underline{\underline{m}}, t = 1$



DS Lower Bounds

- Two trivial solutions:
  - $s \cancel{= n}, t = 1$
  - $s = n, t \cancel{=} n$



DS Lower Bounds

- Two trivial solutions:
  - $\underline{s = m}, t = 1$
  - $s = n, \underline{t = n}$
- There exist problems requiring $\underline{s \approx m}$ or $\underline{t \approx n}$

DS Lower Bounds

- Two trivial solutions:
  - $s = m, t = 1$
  - $s = n, t = n$
- There exist problems requiring $s \approx m$ or $t \approx n$
- Best known concrete lower bound [Sie89]:

$$t \geq \Omega\left(\frac{\log m}{\log(s/n)}\right)$$

$m = poly(n)$

$\log m = O(\log n)$

DS LOWER BOUNDS

- Two trivial solutions:
  - $s = m, t = 1$
  - $s = n, t = n$
- There exist problems requiring $\underline{s} \approx m$ or $t \approx \underline{n}$
- Best known concrete lower bound [Sie89]:

$$t \geq \Omega\left(\frac{\log m}{\log(s/n)}\right)$$

- $\boxed{s = O(n)} \implies \underline{t \geq \Omega(\log n)}$

we have an explicit problem (say, ECC) s.t. every DS that uses linear space $s = O(n)$ must have $t \geq \Omega(\log n)$

DS LOWER BOUNDS

- Two trivial solutions:
    - $s = m, t = 1$
    - $s = n, t = n$
- There exist problems requiring $s \approx m$ or $t \approx n$
- Best known concrete lower bound [Sie89]:

$$t \geq \Omega\left(\frac{\log m}{\log(s/n)}\right)$$

- $\underline{s = O(n)} \implies \underline{t \geq \Omega(\log n)}$
- $s = n^{1+\varepsilon} \implies t \geq \Omega(1)$ — trivial

CIRCUIT LOWER BOUNDS

- n inputs, $m = O(n)$ outputs

$f : \{0,1\}^n \rightarrow \{0,1\}^m$

$m = O(n)$

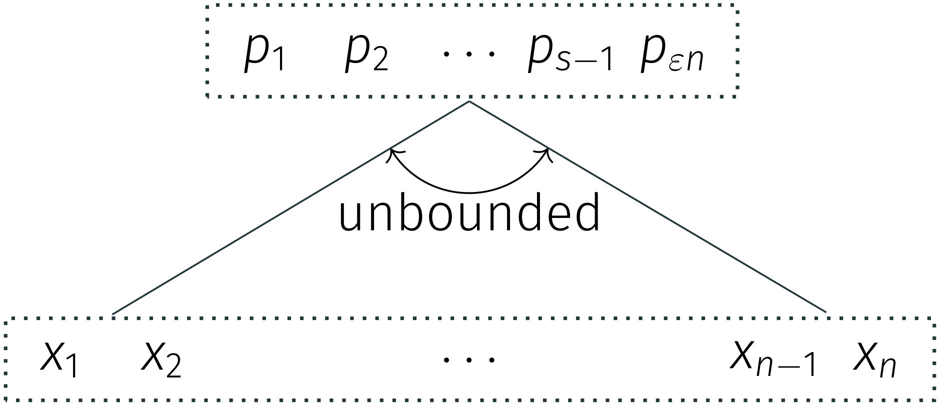
Circuit Lower Bounds

- n inputs, $m = O(n)$ outputs
- $O(n)$ size, $O(\log n)$ depth

CIRCUITS PICTORIALLY

$x_1 \quad x_2 \quad \cdots \quad x_{n-1} \quad x_n$

Circuits Pictorially

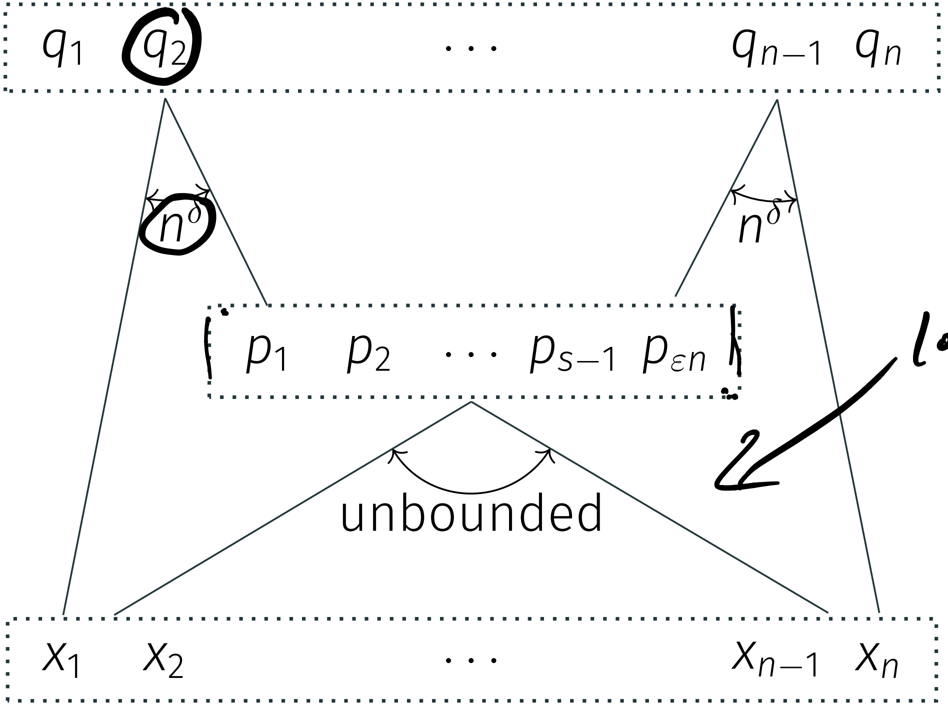


$p_1 \quad p_2 \quad \cdots \quad p_{s-1} \quad p_{\varepsilon n}$

unbounded

$x_1 \quad x_2 \quad \cdots \quad x_{n-1} \quad x_n$

CIRCUITS PICTORIALLY



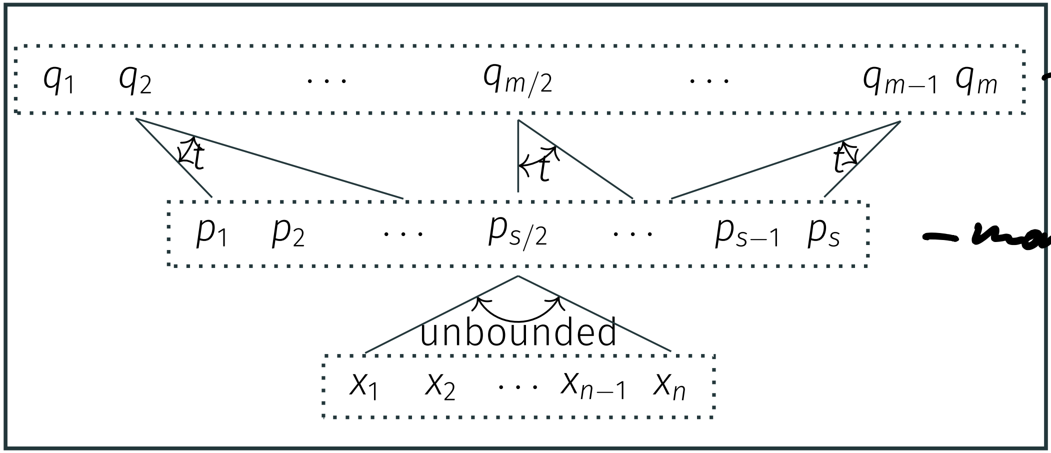
$0 < \varepsilon, \delta < 1$

lose information

If we have $f: \{0,1\}^n \to \{0,1\}^n$ that cannot be computed by such a depth-2 circuit $\overset{[Val77]}{\Longrightarrow}$ f requires $\omega(n)$-size ckts of $O(\log n)$-depth

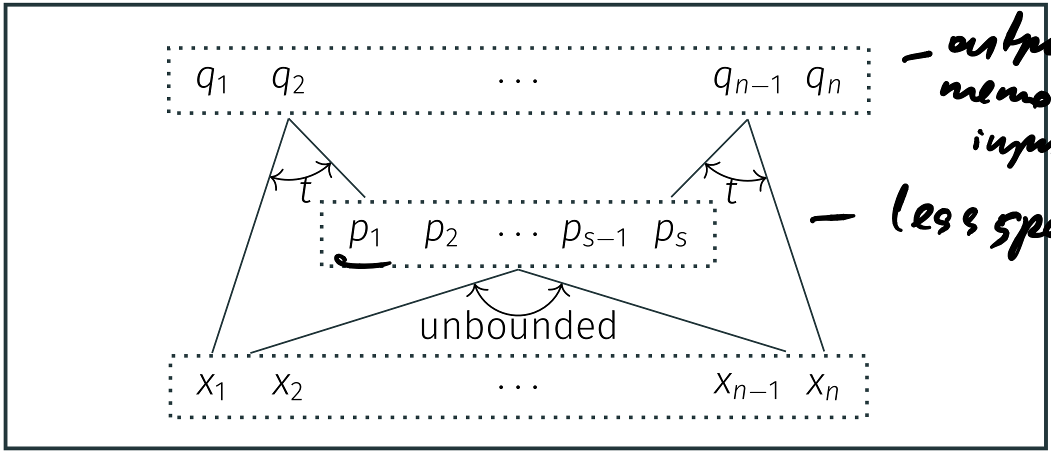
COMPARISON

DS



- outputs look only at memory cells
- more space

Ckts



- outputs look at memory cells & inputs
- less space

Linear Problems

LINEAR CIRCUITS

- A linear circuit computes Mx for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$

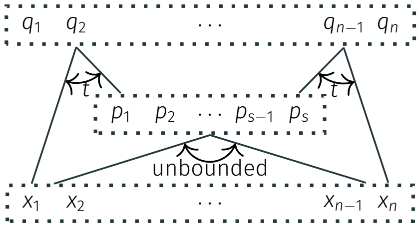
$x \rightarrow Mx$

LINEAR CIRCUITS

- A linear circuit computes Mx for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$
- For a circuit of size $O(n)$ and depth $O(\log n)$,

LINEAR CIRCUITS

- A linear circuit computes Mx for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$
- For a circuit of size $O(n)$ and depth $O(\log n)$,



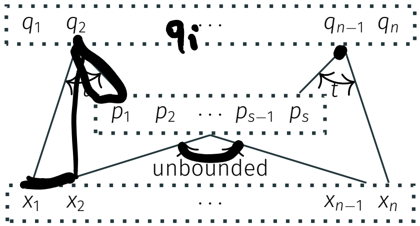
LINEAR CIRCUITS

- A linear circuit computes Mx for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$
- For a circuit of size $O(n)$ and depth $O(\log n)$,

memory cells on inputs

outputs depend on memory cells

$$M = A + \boxed{C} \cdot \boxed{D}$$



$x \to Mx$

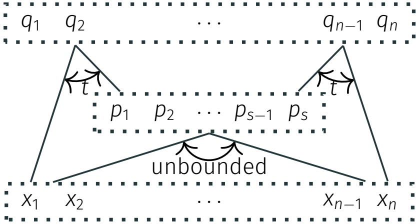
M, i^{th} row describes the dependence of the i^{th} output on the inputs

$\underbrace{A \in \mathbb{F}^{m \times n}}$ describes dependence of outputs on inputs

LINEAR CIRCUITS

- A linear circuit computes Mx for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m\times n}$
- For a circuit of size $O(n)$ and depth $O(\log n)$,

$$\overset{m \times n}{M} = \overset{m \times n}{A} + \overset{m \times \varepsilon n}{C} \cdot \overset{\varepsilon n \times n}{D}$$



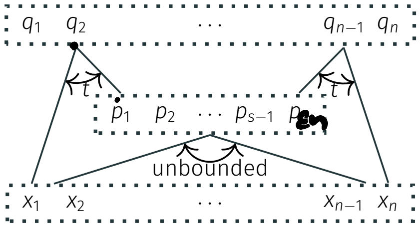
LINEAR CIRCUITS

- A linear circuit computes Mx for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m\times n}$
- For a circuit of size $O(n)$ and depth $O(\log n)$,

$$\overset{m \times n}{M} = \underline{\overset{m \times n}{A}} + \underline{\overset{m \times \varepsilon n}{C}} \cdot \overset{\varepsilon n \times n}{D}$$

sparse

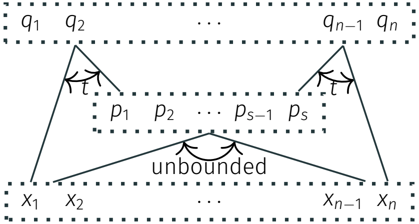
rk(C), rk(D), rk(CD) ≤ εn



LINEAR CIRCUITS

- A linear circuit computes Mx for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$
- For a circuit of size $O(n)$ and depth $O(\log n)$,

$$\overset{m \times n}{M} = \underbrace{\overset{m \times n}{A} + \overset{m \times \varepsilon n}{C}}_{\text{sparse}} \cdot \overset{\varepsilon n \times n}{D} = \underset{\text{sparse}}{A} + \underset{\text{low-rank}}{B}$$



LINEAR CIRCUITS

- A linear circuit computes Mx for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m\times n}$

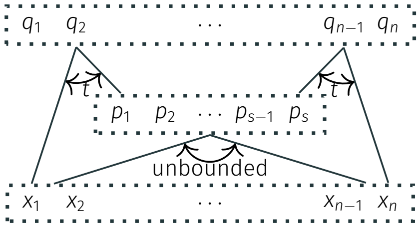
[Val77]

- For a circuit of size $O(n)$ and depth $O(\log n)$,

$$\overset{m\times n}{M} = \overset{m\times n}{A} + \overset{m\times \varepsilon n}{C} \cdot \overset{\varepsilon n\times n}{D} = A + B$$

sparse sparse

low-rank

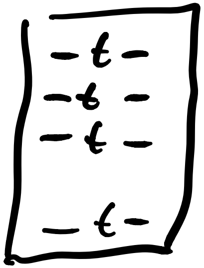


Today, t-sparse means ≤ t non-zeros in every row

- $M \in \mathbb{F}^{m\times n}$ is $(\varepsilon n, t)$-rigid iff

Row-sparsity

$$M \neq \underset{t\text{-sparse}}{A} + \underset{\mathrm{rk} \leq \varepsilon n}{\boxed{B}}$$



LINEAR DATA STRUCTURES

- A linear DS computes Mx for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$

LINEAR DATA STRUCTURES

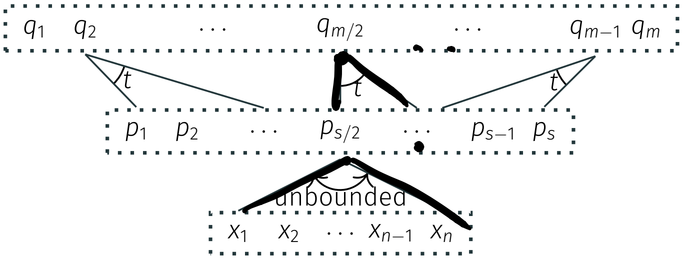
- A linear DS computes Mx for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$

$p_i = Tx, \quad T \in F^{1\times n}$

All memory cells

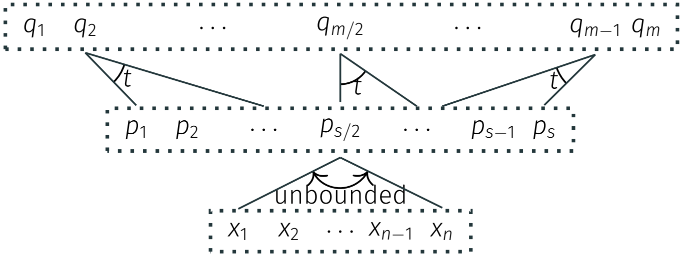
$P = \underline{S} \cdot x, \quad S \in F^{s \times n}$

$x \to Mx$



LINEAR DATA STRUCTURES

- A linear DS computes Mx for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$



$$M = A \cdot B$$

LINEAR DATA STRUCTURES

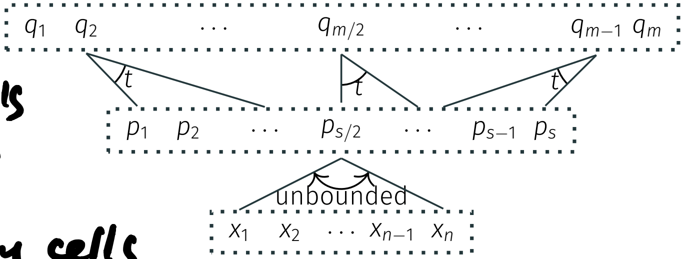
- A linear DS computes Mx for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$

$$\overset{m \times n}{M} = \overset{m \times s}{A \cdot} \overset{s \times n}{B}$$

dep of memory cells on inputs

dep of outputs on memory cells

t-sparse



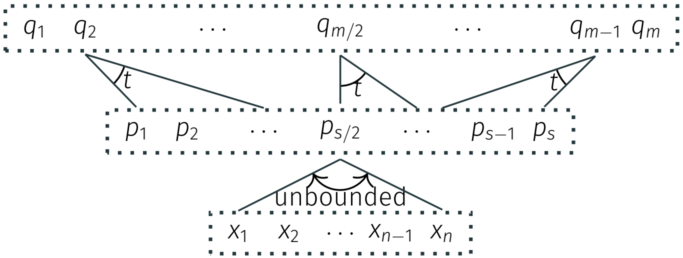
LINEAR DATA STRUCTURES

$m = n^{100}$

$s = n \text{ polylog} n$

- A linear DS computes Mx for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$

$$\overset{m \times n}{M} = \underbrace{\overset{m \times s}{A}}_{t\text{-sparse}} \cdot \underbrace{\overset{s \times n}{B}}_{\text{small}}$$



M is an $\underline{n^{100} \times n}$

B is an $\underline{n \text{polylog} n \times n}$ matrix

Comparison

Small circuit / Non-rigid

$$\overset{m \times n}{M} = \underset{t\text{-sparse}}{\overset{m \times n}{A}} + \underset{\mathbf{rk} \le \varepsilon n}{\overset{m \times n}{B}}$$

Comparison

Small circuit / Non-rigid

$x \to Mx$

$$\overset{m \times n}{M} = \underbrace{\overset{m \times n}{A}}_{t\text{-sparse}} + \underbrace{\overset{m \times n}{B}}_{\mathbf{rk} \leq \varepsilon n}$$

Efficient Data Structure

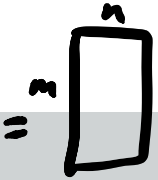
$$\overset{m \times n}{M} = \underbrace{\overset{m \times s}{A}}_{t\text{-sparse}} \cdot \underbrace{\overset{s \times n}{B}}_{\text{small}}$$

DS and Rigidity

RIGIDITY LEMMA

Lemma $\mathrm{Rk}(M)=n$

$M =$



If $M \in \mathbb{F}^{\underline{m\times n}}$ *is not* $(\varepsilon n, t)$*-rigid, then* $\exists A \in \mathbb{F}^{\underline{m\times n}}$ *s.t.* A *is* $\underline{\text{t-sparse}}$*, and*

$$\dim(\underline{\mathrm{Col}(A)} \cap \underline{\mathrm{Col}(M)}) \geq \underline{n(1-2\varepsilon)}\,.$$

RIGIDITY LEMMA

Lemma

If $M \in \mathbb{F}^{m \times n}$ is not $(\varepsilon n, t)$-rigid, then $\exists A \in \mathbb{F}^{m \times n}$ s.t. A is t-sparse, and

$$\dim(\mathrm{Col}(A) \cap \mathrm{Col}(M)) \geq n(1 - 2\varepsilon)\,.$$

Proof.

$$M = \underset{t\text{-sparse}}{A} + \underset{\mathrm{rk} \leq \varepsilon n}{B}$$

RIGIDITY LEMMA

Lemma

If $M \in \mathbb{F}^{m\times n}$ is not $(\varepsilon n, t)$-rigid, then $\exists A \in \mathbb{F}^{m\times n}$ s.t. A is t-sparse, and

$$\dim(\mathrm{Col}(A) \cap \mathrm{Col}(M)) \geq n(1-2\varepsilon)\,.$$

Proof.

$$\boxed{M} = \underset{t\text{-sparse}}{A} + \underset{\mathrm{rk} \leq \varepsilon n}{B} \qquad L \in \mathbb{F}^{m\times m}, \underline{\ker(L)} = \underline{\mathrm{Col}(M)}$$

$U = \underline{\mathrm{Col}(M)}$, $\underline{U} \subseteq \mathbb{F}^m$

Def. lin op $L : \mathbb{F}^m \to \mathbb{F}^m$

$L(U) = 0$ ✓

$\underline{L(x) \neq 0}$ if $x \notin U$

$e_1 \ldots e_n$ - $\underline{\text{basis of } U}$

$e_1 \ldots e_n \ldots e_m$ - basis of $\mathbb{F}^m$

$L(e_1) = \ldots = \underline{L(e_n) = 0}$

$\boxed{L(e_i) = e_i}$ $i > n$

RIGIDITY LEMMA

Lemma

If $M \in \mathbb{F}^{m\times n}$ is not $(\varepsilon n, t)$-rigid, then $\exists A \in \mathbb{F}^{m\times n}$ s.t. A is t-sparse, and

$$\dim(\mathrm{Col}(A) \cap \mathrm{Col}(M)) \geq n(1-2\varepsilon)\,.$$

Proof.

✓ $$M = \underbrace{A}_{t\text{-sparse}} + \underbrace{B}_{\mathrm{rk} \leq \varepsilon n}$$

$$L \in \mathbb{F}^{m\times m}, \boxed{\ker(L)} = \boxed{\mathrm{Col}(M)}$$

$$0 = LA + LB \qquad LA = -LB$$

$$LM = LA + LB$$

RIGIDITY LEMMA

Lemma

If $M \in \mathbb{F}^{m\times n}$ is not $(\varepsilon n, t)$-rigid, then $\exists A \in \mathbb{F}^{m\times n}$ s.t. A is t-sparse, and

$$\dim(\mathrm{Col}(A) \cap \mathrm{Col}(M)) \geq n(1-2\varepsilon)\,.$$

Proof.

$$M = \underbrace{A}_{t\text{-sparse}} + \underbrace{B}_{\underline{\mathsf{rk} \leq \varepsilon n}} \qquad L \in \mathbb{F}^{m\times m}, \mathsf{ker}(L) = \mathrm{Col}(M)$$

$$LM = LA + LB \implies \underline{\mathsf{rk}(LA)} = \underline{\mathsf{rk}(LB)}$$

RIGIDITY LEMMA

Lemma

If $M \in \mathbb{F}^{m \times n}$ is not $(\varepsilon n, t)$-rigid, then $\exists A \in \mathbb{F}^{m \times n}$ s.t. A is t-sparse, and

$$\dim(\mathrm{Col}(A) \cap \mathrm{Col}(M)) \geq n(1 - 2\varepsilon)\,.$$

Proof.

$$M = \underbrace{A}_{t\text{-sparse}} + \underbrace{B}_{\mathrm{rk} \leq \varepsilon n} \qquad L \in \mathbb{F}^{m \times m}, \ker(L) = \mathrm{Col}(M)$$

$$LM = LA + LB \implies \mathrm{rk}(LA) = \underline{\mathrm{rk}(LB)} \ \underline{\leq \varepsilon n}$$

Rigidity Lemma

Lemma $\underline{\mathrm{rk}(M)=n}$

If $M \in \mathbb{F}^{m\times n}$ is not $(\varepsilon n, t)$-rigid, then $\exists A \in \mathbb{F}^{m\times n}$ s.t. A is t-sparse, and

$$\dim(\mathrm{Col}(A) \cap \mathrm{Col}(M)) \geq n(1-2\varepsilon)\,.$$

Proof.

$$\boxed{M = \underbrace{A}_{t\text{-sparse}} + \underbrace{B}_{\mathrm{rk} \leq \underline{\varepsilon n}}} \qquad L \in \mathbb{F}^{m\times m}, \ker(L) = \mathrm{Col}(M)$$

$$LM = LA + LB \implies \mathrm{rk}(LA) = \mathrm{rk}(LB) \leq \underline{\varepsilon n}$$

$$A = M - B$$

$$\mathrm{rk}(A) \geq \underline{n - \varepsilon n}$$

rk(A) is high ; rk(LA) is low

RIGIDITY LEMMA

Lemma

If $M \in \mathbb{F}^{m \times n}$ is not $(\varepsilon n, t)$-rigid, then $\exists A \in \mathbb{F}^{m \times n}$ s.t. A is $\underline{t\text{-sparse}}$, and

$$\dim(\underline{\mathrm{Col}(A)} \cap \underline{\mathrm{Col}(M)}) \geq n(1 - 2\varepsilon)\,.$$

Proof.

Found A

$$M = \underbrace{A}_{\boxed{t\text{-sparse}}} + \underbrace{B}_{\mathrm{rk} \leq \varepsilon n} \qquad L \in \mathbb{F}^{m \times m}, \ker(L) = \boxed{\mathrm{Col}(M)}$$

$$LM = LA + LB \implies \mathrm{rk}(LA) = \mathrm{rk}(LB) \leq \underline{\varepsilon n}$$

$$\mathrm{rk}(A) \geq \underline{n - \varepsilon n} \implies \dim(\boxed{\mathrm{Col}(A)} \cap \boxed{\ker(L)}) \geq \underline{n(1 - 2\varepsilon)} \qquad \square$$

DS and Rigidity

Theorem

Every $M \in \mathbb{F}^{m\times n}$

- ✗ *either is computable by a DS with* $s = \underline{(1+\varepsilon)n}$ *and* $\underline{\underline{t}}$,
- ✓ *or it contains a submatrix* $M' \in \mathbb{F}^{m\times n'}$ *which is* $\underline{(\varepsilon n', \frac{t}{\log n})\text{-rigid}}$.

DS LB for M => rigid M'

DS LB => rigidity

DS LB => Ckt LB

Later Ckt LB => DS LB

DS AND RIGIDITY

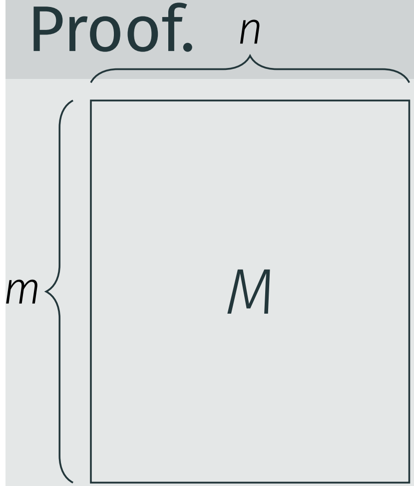
Theorem

Every $M \in \mathbb{F}^{m \times n}$

- *either is computable by a DS with* $s = (1 + \varepsilon)n$ *and* t,
- *or it contains a submatrix* $M' \in \mathbb{F}^{m \times n'}$ *which is* $(\varepsilon n', \frac{t}{\log n})$*-rigid.*

Proof.

If M is rigid, then we're done
So assume M is not rigid



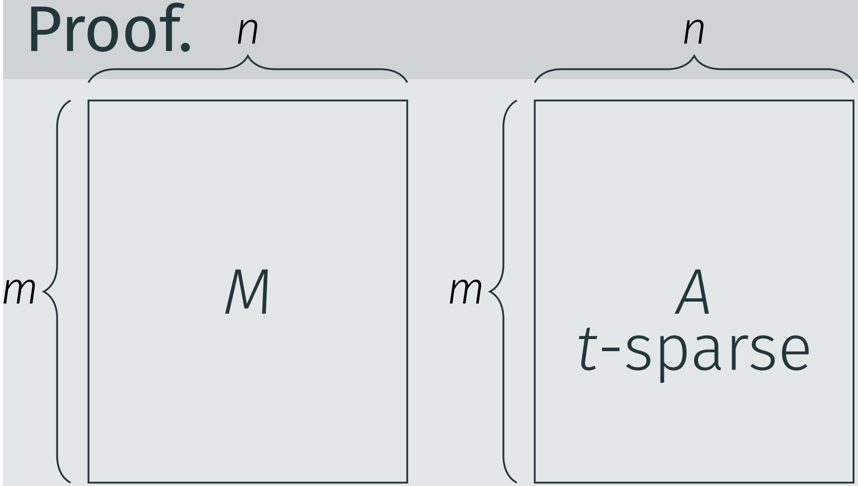
DS AND RIGIDITY

Theorem

Every $M \in \mathbb{F}^{m\times n}$

- *either is computable by a DS with* $s = (1+\varepsilon)n$ *and* t,
- *or it contains a submatrix* $M' \in \mathbb{F}^{m\times n'}$ *which is* $(\varepsilon n', \frac{t}{\log n})$*-rigid.*

Proof.



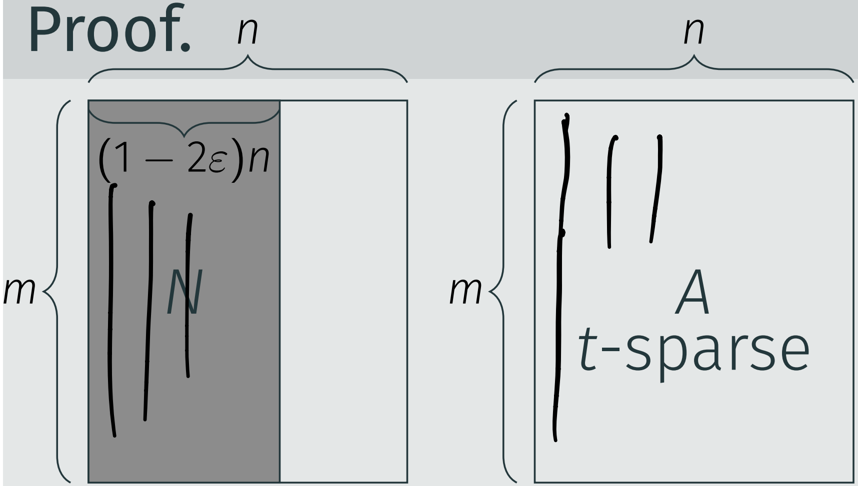
DS and Rigidity

Theorem

Every $M \in \mathbb{F}^{m\times n}$

- *either is computable by a DS with* $s = (1+\varepsilon)n$ *and* t,
- *or it contains a submatrix* $M' \in \mathbb{F}^{m\times n'}$ *which is* $(\varepsilon n', \frac{t}{\log n})$*-rigid.*

Proof.



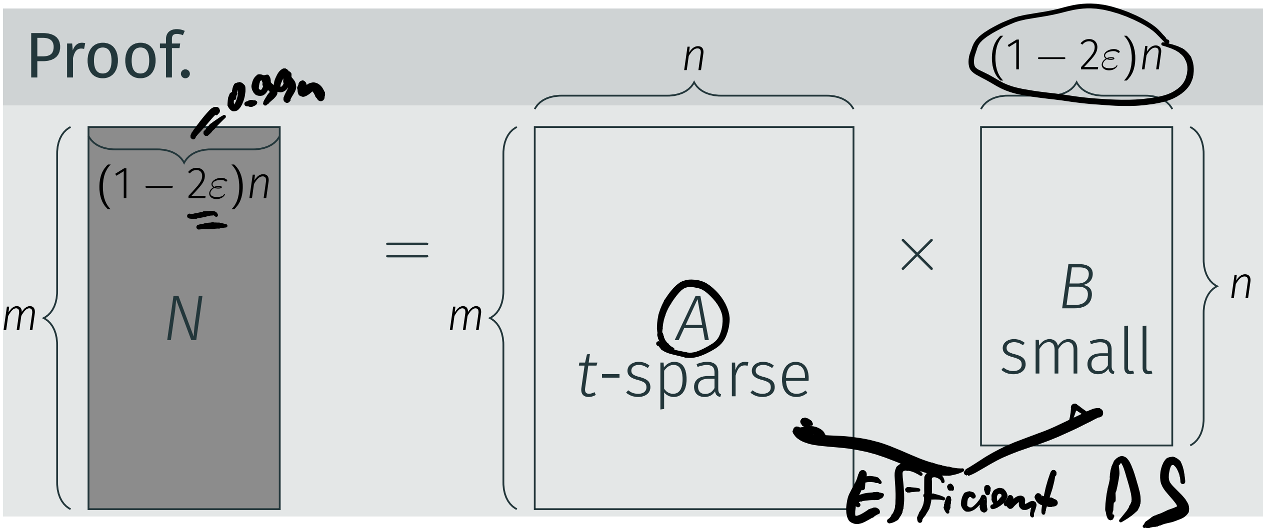
DS AND RIGIDITY

Theorem

Every $M \in \mathbb{F}^{m \times n}$

- *either is computable by a DS with* $s = (1+\varepsilon)n$ *and* t,
- *or it contains a submatrix* $M' \in \mathbb{F}^{m \times n'}$ *which is* $(\varepsilon n', \frac{t}{\log n})$*-rigid.*

Proof.



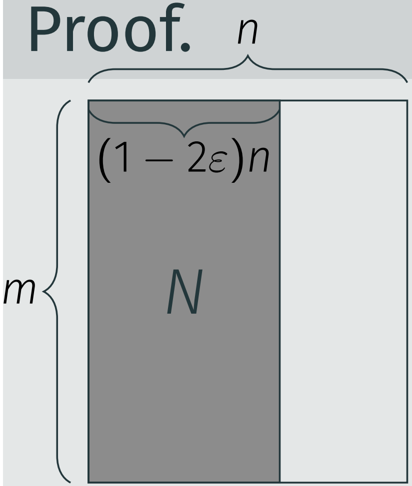
DS AND RIGIDITY

Theorem

Every $M \in \mathbb{F}^{m \times n}$

- *either is computable by a DS with* $s = (1+\varepsilon)n$ *and* t,
- *or it contains a submatrix* $M' \in \mathbb{F}^{m \times n'}$ *which is* $(\varepsilon n', \frac{t}{\log n})$*-rigid.*

Proof.



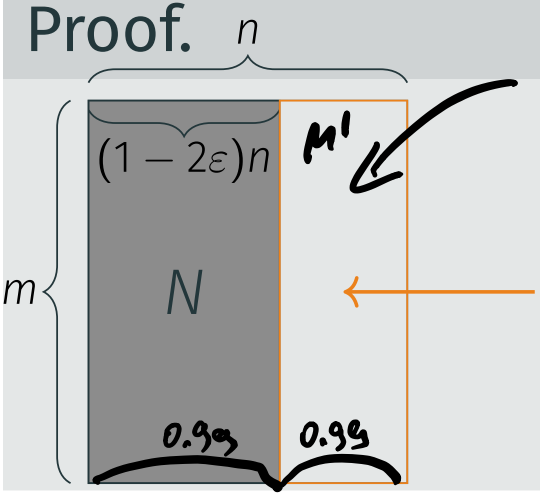
DS and Rigidity

Theorem $\Rightarrow$ DSLB $\Rightarrow$ Cat LB

Every $M \in \mathbb{F}^{m\times n}$

- *either is computable by a DS with* $s = (1+\varepsilon)n$ *and* t,
- *or it contains a submatrix* $M' \in \mathbb{F}^{m\times n'}$ *which is* $(\varepsilon n', \frac{t}{\boxed{\log n}})$*-rigid.*

Proof.



Recurse

IF M' is rigid — we're done

Otherwise, we'll find an efficient DS for 99% of M'

After log n steps I'll have eff DSs for all parts of M

□

Equivalence

Paturi-Pudlák Dimensions

- $M \in \mathbb{F}^{m\times n}, \underline{t \in \mathbb{N}.}$

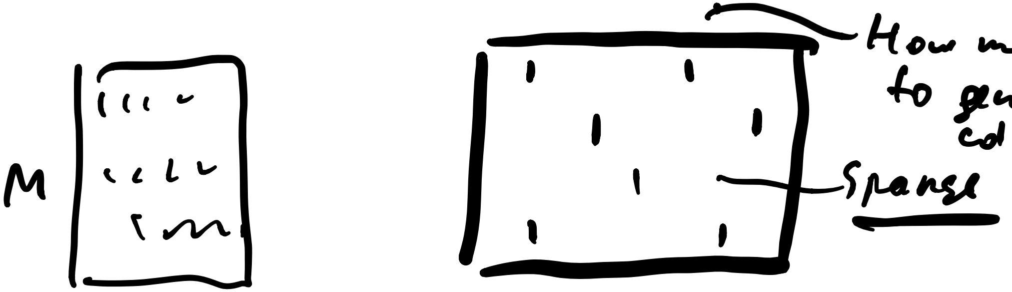
PATURI-PUDLÁK DIMENSIONS

- $\underline{M} \in \mathbb{F}^{m \times n}, t \in \mathbb{N}$.
- Outer dimension

min # of cols in a sparse matrix generating M.

$$D_M(\boxed{t}) = \min\{s : \exists A \in \mathbb{F}^{m \times s},$$

$$A \text{ is } t\text{-sparse}, \mathrm{Col}(M) \subseteq \mathrm{Col}(A)\}.$$

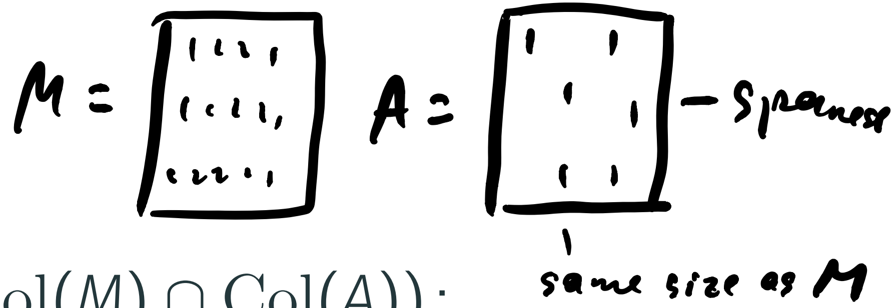


How many cols I need to generate same col space?

Paturi-Pudlák Dimensions

- $M \in \mathbb{F}^{m \times n}, t \in \mathbb{N}.$
- Outer dimension

$$D_M(t) = \min\{s \colon \exists A \in \mathbb{F}^{m \times s}, \; A \text{ is } t\text{-sparse}, \mathrm{Col}(M) \subseteq \mathrm{Col}(A)\}\,.$$



- Inner dimension

$$d_M(t) = \max\{\underline{\dim(\mathrm{Col}(M) \cap \mathrm{Col}(A))} \colon A \in \mathbb{F}^{m \times n}, A \text{ is } t\text{-sparse}\}\,.$$

PATURI-PUDLÁK DIMENSIONS

Linear problems, DSs, ckts

$\underline{M \in \mathbb{F}^{m \times n}}$ is computable by (s, t) DS

$$\Longleftrightarrow$$

$$\underline{D_M(t) \leq} \, s \, .$$

small DS ⟺ small out. dim

PATURI-PUDLÁK DIMENSIONS

$M \in \mathbb{F}^{m \times n}$ is computable by (s, t) DS

✓ $$\iff$$

$$D_M(t) \leq s\,.$$

$M \in \mathbb{F}^{m \times n}$ is $(\varepsilon n, t)$-strongly rigid

✓ $$\iff$$

$$d_M(t) \leq n - \varepsilon n\,.$$

Rigid matrix ≡
≡ small inner dim

PP DIM INEQUALITIES

Theorem

$$\underline{D_M(t) \geq (1+\varepsilon)n} \implies \underline{d_{M'}\left(\frac{t}{\log n}\right)} \leq n(1-\varepsilon)\,.$$

DS LB => rigidity LB

PP DIM INEQUALITIES

Theorem

$$D_M(t) \geq (1+\varepsilon)n \implies d_{M'}\left(\frac{t}{\log n}\right) \leq n(1-\varepsilon)\,.$$

DS LB => Rigidity LB

Lemma

$$\underline{D_M}(t) + \underline{d_M(t)} \geq 2n\,.$$

If $D \leq n-k$ =>

$d \geq n+k$

Rigidity LB => DS LB

PP Dim Inequalities

Theorem

$$D_M(t) \geq (1+\varepsilon)n \implies d_{M'}\left(\frac{t}{\log n}\right) \leq n(1-\varepsilon)\,.$$

Lemma

$$D_M(t) + d_M(t) \geq 2n\,.$$

Lemma

$M \in \mathbb{F}^{O(n)\times n}$, $\underline{D_M(n^\delta)} \geq (1+\varepsilon)n \implies$
M doesn't have $O(n)$ size $O(\log n)$ depth ckts.

DS IMPLIES RIGIDITY

Theorem

$$D_M(t) \geq (1+\varepsilon)n \implies d_{M'}\left(\frac{t}{\log n}\right) \leq n(1-\varepsilon)\,.$$

DS IMPLIES RIGIDITY

Theorem

$$D_M(t) \geq (1+\varepsilon)n \implies d_{M'}\left(\frac{t}{\log n}\right) \leq n(1-\varepsilon)\,.$$

Corollary

- *Any DS lower bound of* $t \geq \omega(\log^2 n)$ *for* $s =$ $(1+\varepsilon)n$ *would imply a new rigidity bound for rectangular matrices.*

DS IMPLIES RIGIDITY

Theorem

$$D_M(t) \geq (1+\varepsilon)n \implies d_{M'}\left(\frac{t}{\log n}\right) \leq n(1-\varepsilon)\,.$$

Corollary

- *Any DS lower bound of* $t \geq \omega(\log^2 n)$ *for* $s = (1+\varepsilon)n$ *would imply a new rigidity bound for rectangular matrices.*
- *Any DS lower bound of* $t \geq \Omega(\log^{3+\varepsilon} n)$ *for* $s = (1+\varepsilon)n$ *would imply a new rigidity bound for square matrices with* $r = o(n)$*.*

RIGIDITY IMPLIES DS

Lemma

$$D_M(t) + d_M(t) \geq 2n\,.$$

RIGIDITY IMPLIES DS

Lemma

$$D_M(t) + d_M(t) \geq 2n\,.$$

Proof.

$M \in \mathbb{F}^{m\times n}$

RIGIDITY IMPLIES DS

Lemma

$$D_M(t) + d_M(t) \geq 2n\,.$$

Proof.

$M \in \mathbb{F}^{m\times n}$ $\qquad$ $A \in \mathbb{F}^{m\times D_M(t)}, \mathrm{Col}(M) \subseteq \mathrm{Col}(A)$

RIGIDITY IMPLIES DS

Lemma

$$D_M(t) + d_M(t) \geq 2n\,.$$

Proof.

$M \in \mathbb{F}^{m \times n}$ $\qquad$ $A \in \mathbb{F}^{m \times D_M(t)}, \mathrm{Col}(M) \subseteq \mathrm{Col}(A)$

$B \in \mathbb{F}^{m \times n}$ - the first n columns of A

RIGIDITY IMPLIES DS

Lemma

$$D_M(t) + d_M(t) \geq 2n\,.$$

Proof.

$M \in \mathbb{F}^{m \times n}$ $\quad$ $A \in \mathbb{F}^{m \times D_M(t)}, \mathrm{Col}(M) \subseteq \mathrm{Col}(A)$

$B \in \mathbb{F}^{m \times n}$ - the first n columns of A

$$D_M(t) = \dim(\mathrm{Col}(A)) \geq \dim(\mathrm{Col}(B) + \mathrm{Col}(M))$$

RIGIDITY IMPLIES DS

Lemma

$$D_M(t) + d_M(t) \geq 2n\,.$$

Proof.

$M \in \mathbb{F}^{m\times n}$ $\qquad$ $A \in \mathbb{F}^{m\times D_M(t)}, \mathrm{Col}(M) \subseteq \mathrm{Col}(A)$

$B \in \mathbb{F}^{m\times n}$ - the first n columns of A

$$D_M(t) = \mathbf{dim}(\mathrm{Col}(A)) \geq \mathbf{dim}(\mathrm{Col}(B) + \mathrm{Col}(M))$$

$$= \mathbf{dim}(\mathrm{Col}(B)) + \mathbf{dim}(\mathrm{Col}(M)) - \mathbf{dim}(\mathrm{Col}(B) \cap \mathrm{Col}(M))$$

RIGIDITY IMPLIES DS

Lemma

$$D_M(t) + d_M(t) \geq 2n\,.$$

Proof.

$M \in \mathbb{F}^{m\times n}$ $\qquad$ $A \in \mathbb{F}^{m\times D_M(t)}, \mathrm{Col}(M) \subseteq \mathrm{Col}(A)$

$B \in \mathbb{F}^{m\times n}$ - the first n columns of A

$$D_M(t) = \mathbf{dim}(\mathrm{Col}(A)) \geq \mathbf{dim}(\mathrm{Col}(B) + \mathrm{Col}(M))$$

$$= \mathbf{dim}(\mathrm{Col}(B)) + \mathbf{dim}(\mathrm{Col}(M)) - \mathbf{dim}(\mathrm{Col}(B) \cap \mathrm{Col}(M))$$

$$= 2n - d_M(t)\,.$$

□

RIGIDITY IMPLIES DS

Lemma

$$D_M(t) + d_M(t) \geq 2n\,.$$

Corollary

An $(\varepsilon n, t)$*-strongly rigid matrix would imply a DS lower bound of* $s = (1+\varepsilon)n$ *and* t*.*

STRONG DS LB IMPLY CIRCUIT LB

Lemma

$M \in \mathbb{F}^{O(n)\times n}, D_M(n^\delta) \geq (1+\varepsilon)n \implies$
M doesn't have $O(n)$ *size* $O(\log n)$ *depth ckts.*

STRONG DS LB IMPLY CIRCUIT LB

Lemma

$M \in \mathbb{F}^{O(n)\times n}, D_M(n^\delta) \geq (1+\varepsilon)n \implies$ M doesn't have $O(n)$ size $O(\log n)$ depth ckts.

Proof.

Circuit of size $O(n)$ and depth $O(\log n)$,

STRONG DS LB IMPLY CIRCUIT LB

Lemma

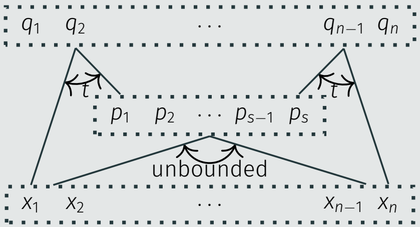
*$M \in \mathbb{F}^{O(n)\times n}, D_M(n^\delta) \geq (1+\varepsilon)n \implies$
M doesn't have $O(n)$ size $O(\log n)$ depth ckts.*

Proof.

Circuit of size $O(n)$ and depth $O(\log n)$,

$$\overset{m \times n}{M} = \overset{m \times n}{A} + \overset{m \times \varepsilon n}{C} \cdot \overset{\varepsilon n \times n}{D}$$

sparse



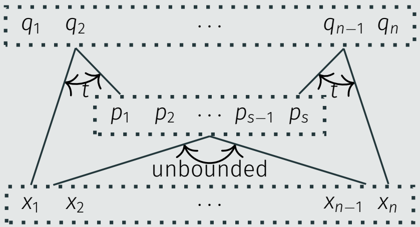
Strong DS LB Imply Circuit LB

Lemma

$M \in \mathbb{F}^{O(n)\times n}, D_M(n^\delta) \geq (1+\varepsilon)n \implies$
M doesn't have $O(n)$ size $O(\log n)$ depth ckts.

Proof.

Circuit of size $O(n)$ and depth $O(\log n)$,



$$\overset{m\times n}{M} = \underbrace{\overset{m\times n}{A} + \overset{m\times \varepsilon n}{C}}_{\text{sparse}} \cdot \overset{\varepsilon n\times n}{D}$$

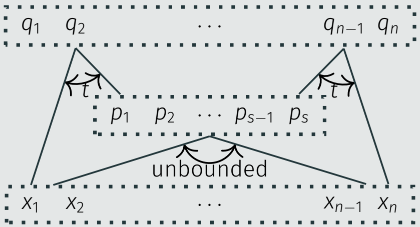
$$\mathrm{Col}(M) \subseteq \mathrm{Col}(A) \cup \mathrm{Col}(C)$$

Strong DS LB Imply Circuit LB

Lemma

$M \in \mathbb{F}^{O(n)\times n}, D_M(n^\delta) \geq (1+\varepsilon)n \implies$
M doesn't have $O(n)$ size $O(\log n)$ depth ckts.

Proof.



Circuit of size $O(n)$ and depth $O(\log n)$,

$$\overset{m\times n}{M} = \underbrace{\overset{m\times n}{A} + \overset{m\times \varepsilon n}{C}}_{\text{sparse}} \cdot \overset{\varepsilon n\times n}{D}$$

$$\mathrm{Col}(M) \subseteq \mathrm{Col}(A) \cup \mathrm{Col}(C)$$

$$D_M(t) \leq (1+\varepsilon)n\,.$$

$\square$

STRONG DS LB IMPLY CIRCUIT LB

Lemma

$M \in \mathbb{F}^{O(n) \times n}, D_M(n^\delta) \geq (1+\varepsilon)n \implies$ *M doesn't have* $O(n)$ *size* $O(\log n)$ *depth ckts.*

Corollary

A lower bound of $s \approx m$ *for* $t = n^\varepsilon$ *would imply a super-linear circuit lower bound.*

APPLICATIONS

- Any DS lower bound of $t \geq \omega(\log^2 n)$ for $s = (1 + \varepsilon)n$ would imply a new rigidity bound for rectangular matrices.

APPLICATIONS

- Any DS lower bound of $t \geq \omega(\log^2 n)$ for $s = (1+\varepsilon)n$ would imply a new rigidity bound for rectangular matrices.
- Any DS lower bound of $t \geq \Omega(\log^{3+\varepsilon} n)$ for $s = (1+\varepsilon)n$ would imply a new rigidity bound for square matrices with $r = o(n)$.

APPLICATIONS

- Any DS lower bound of $t \geq \omega(\log^2 n)$ for $s = (1+\varepsilon)n$ would imply a new rigidity bound for rectangular matrices.
- Any DS lower bound of $t \geq \Omega(\log^{3+\varepsilon} n)$ for $s = (1+\varepsilon)n$ would imply a new rigidity bound for square matrices with $r = o(n)$.
- An $(\varepsilon n, t)$-strongly rigid matrix would imply a DS lower bound of $s = (1+\varepsilon)n$ and t.

Applications

- Any DS lower bound of $t \geq \omega(\log^2 n)$ for $s = (1+\varepsilon)n$ would imply a new rigidity bound for rectangular matrices.
- Any DS lower bound of $t \geq \Omega(\log^{3+\varepsilon} n)$ for $s = (1+\varepsilon)n$ would imply a new rigidity bound for square matrices with $r = o(n)$.
- An $(\varepsilon n, t)$-strongly rigid matrix would imply a DS lower bound of $s = (1+\varepsilon)n$ and t.
- A DS lower bound of $s \approx m$ for $t = n^{\varepsilon}$ would imply a super-linear circuit lower bound.

DS LB $\Leftarrow\Rightarrow$ Rigidity LB

APPLICATIONS

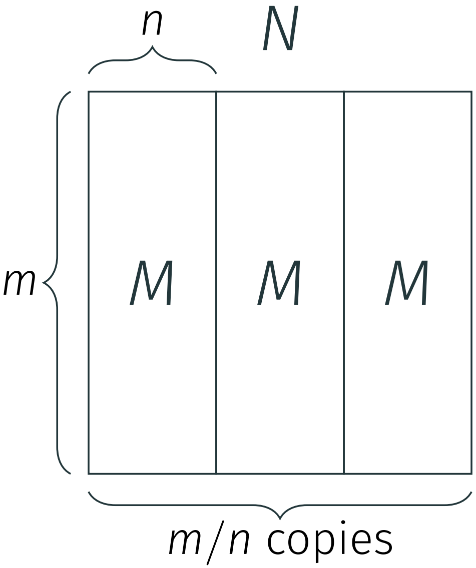
in all regimes of params the best known LB match

- Any DS lower bound of $t \geq \omega(\log^2 n)$ for $s = (1+\varepsilon)n$ would imply a new rigidity bound for rectangular matrices.
- Any DS lower bound of $t \geq \Omega(\log^{3+\varepsilon} n)$ for $s = (1+\varepsilon)n$ would imply a new rigidity bound for square matrices with $r = o(n)$.
- An $(\varepsilon n, t)$-strongly rigid matrix would imply a DS lower bound of $s = (1+\varepsilon)n$ and t.
- A DS lower bound of $s \approx m$ for $t = n^\varepsilon$ would imply a super-linear circuit lower bound.
- Any improvement on rigidity for the regime $r = o(n)$ would lead to a new succinct DS lower bound where $s = n + o(n)$. And vice versa!

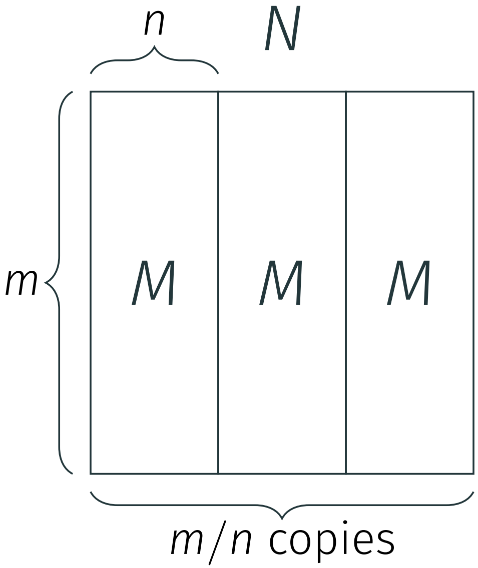
Any improvement for one of these problems => lead to improvement for the other.

RECTANGULAR TO SQUARE RIGIDITY

Rectangular to Square Rigidity



RECTANGULAR TO SQUARE RIGIDITY

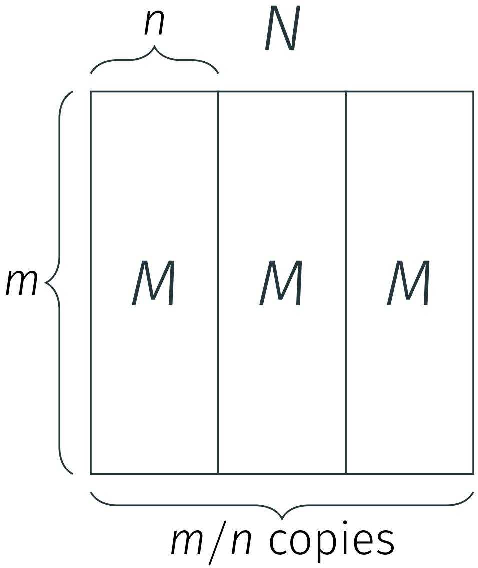


- M is $(\varepsilon n, t)$-rigid

  $\Longrightarrow$

  N is $(\varepsilon n, t \cdot \frac{m}{n})$-rigid ?

RECTANGULAR TO SQUARE RIGIDITY

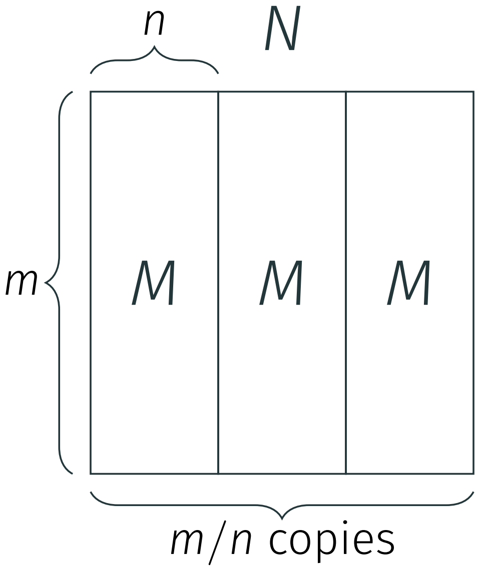


- M is $(\varepsilon n, t)$-rigid

  $\implies$

  N is $(\varepsilon n, t \cdot \frac{m}{n})$-rigid ?
- No!

RECTANGULAR TO SQUARE RIGIDITY

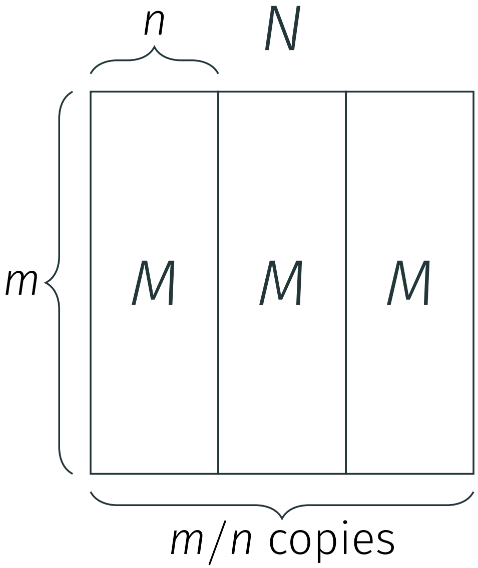


- M is $(\varepsilon n, t)$-rigid

  $\implies$

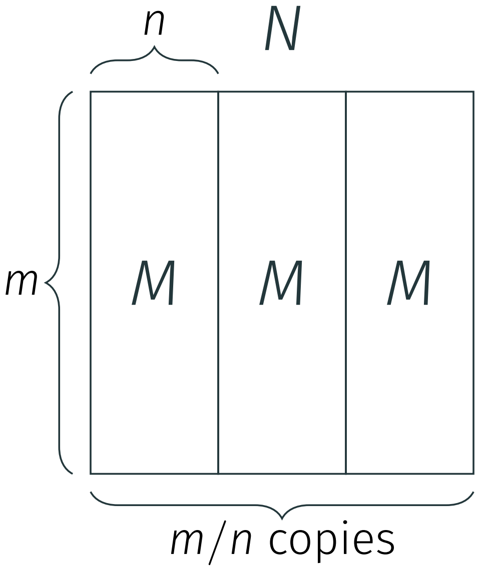
  N is $(\varepsilon n, t \cdot \frac{m}{n})$-rigid ?
- No!
- But it is true for global rigidity

RECTANGULAR TO SQUARE RIGIDITY



- M is $(\varepsilon n, t)$-rigid
  $\Longrightarrow$
  N is $(\varepsilon n, t \cdot \frac{m}{n})$-rigid ?
- No!
- But it is true for global rigidity
- $M \in \mathbb{F}^{m \times n}$ is $(\varepsilon n, s)$-globally rigid

RECTANGULAR TO SQUARE RIGIDITY

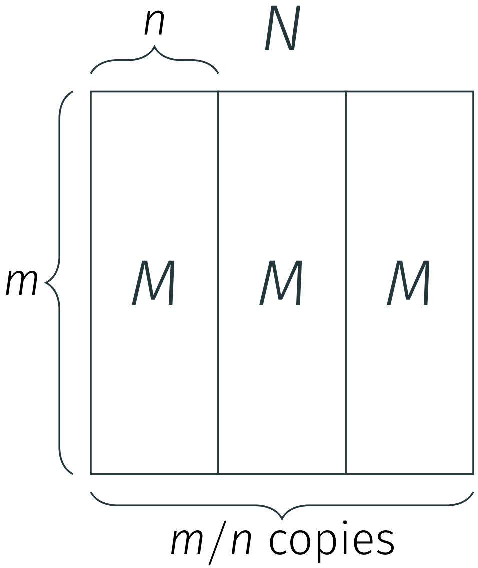


- M is $(\varepsilon n, t)$-rigid

  $\Longrightarrow$

  N is $(\varepsilon n, t \cdot \frac{m}{n})$-rigid ?
- No!
- But it is true for global rigidity
- $M \in \mathbb{F}^{m \times n}$ is $(\varepsilon n, s)$-globally rigid

RECTANGULAR TO SQUARE RIGIDITY



- M is $(\varepsilon n, t)$-rigid

  $\implies$

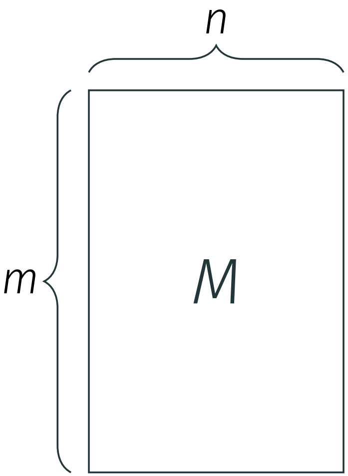
  N is $(\varepsilon n, t \cdot \frac{m}{n})$-rigid ?
- No!
- But it is true for global rigidity
- $M \in \mathbb{F}^{m \times n}$ is $(\varepsilon n, s)$-globally rigid

$$M \neq \underbrace{A}_{\text{has } s \text{ non-zeros}} + \underbrace{B}_{\mathbf{rk} \leq \varepsilon n}$$

ROW TO GLOBAL RIGIDITY

Theorem

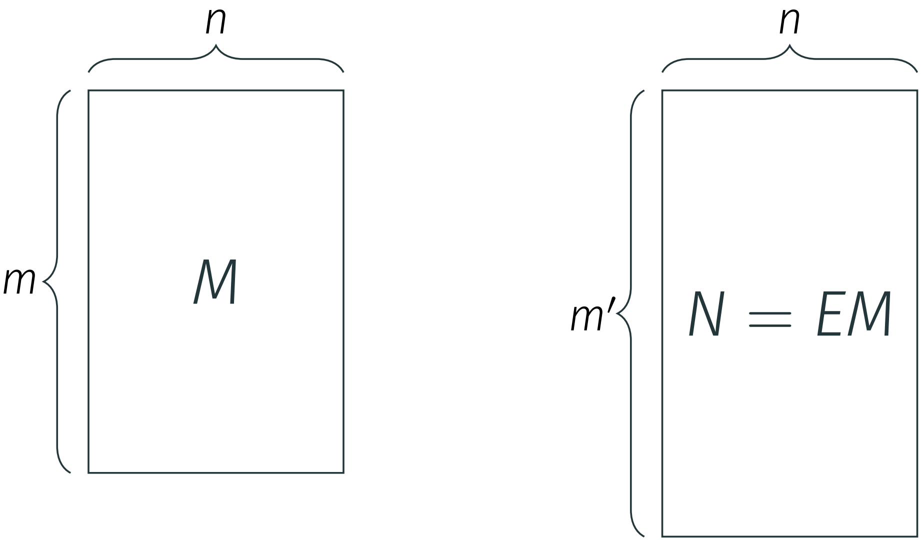
Let $E \in \mathbb{F}^{m' \times m}$ be a q-query LDC. If $M \in \mathbb{F}^{m \times n}$ is (r, t)-rigid, then $N = EM \in \mathbb{F}^{m' \times n}$ is $(r, tm' \cdot \frac{1}{10q})$-globally rigid.



ROW TO GLOBAL RIGIDITY

Theorem

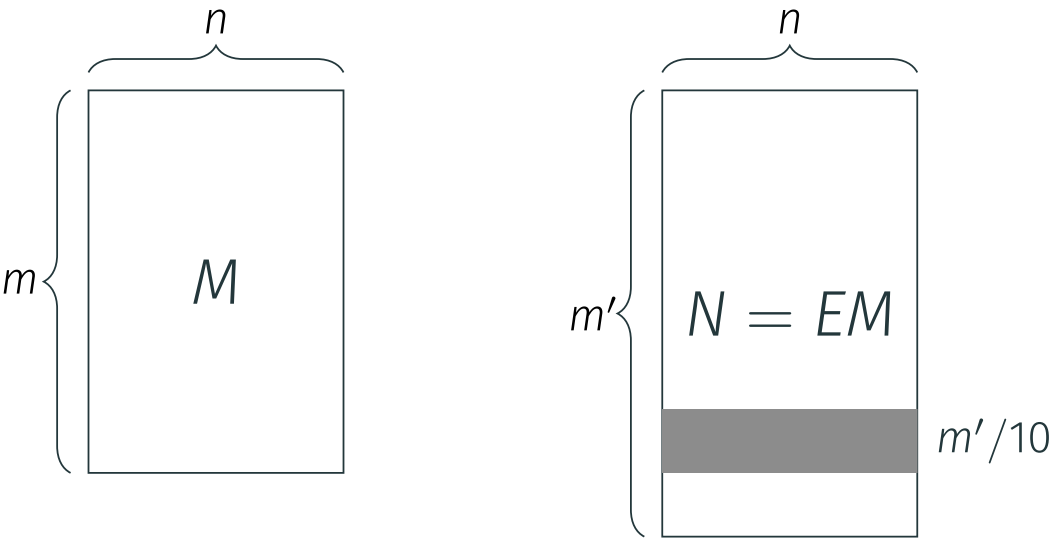
Let $E \in \mathbb{F}^{m' \times m}$ be a q-query LDC. If $M \in \mathbb{F}^{m \times n}$ is (r, t)-rigid, then $N = EM \in \mathbb{F}^{m' \times n}$ is $(r, tm' \cdot \frac{1}{10q})$-globally rigid.



ROW TO GLOBAL RIGIDITY

Theorem

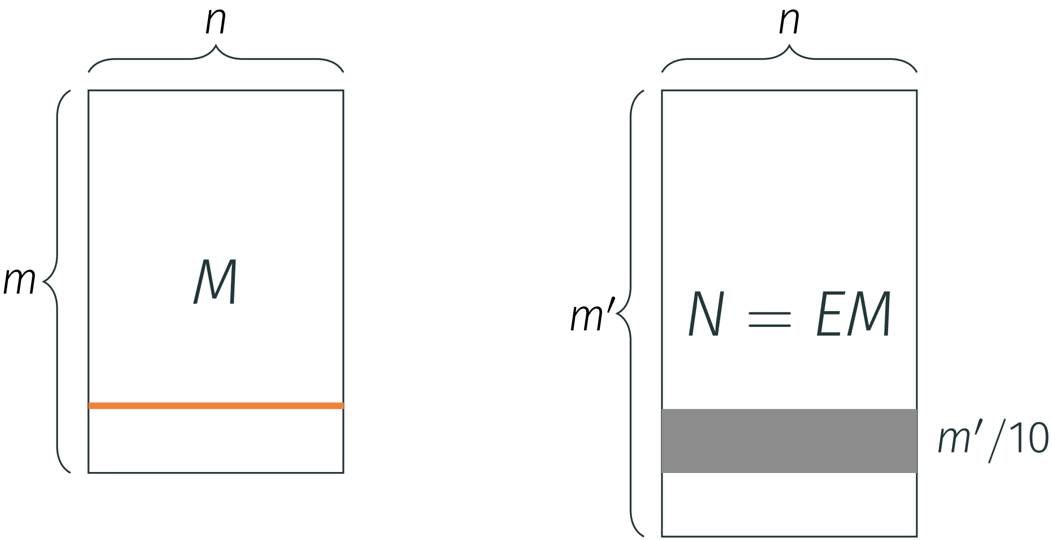
Let $E \in \mathbb{F}^{m' \times m}$ be a q-query LDC. If $M \in \mathbb{F}^{m \times n}$ is (r, t)-rigid, then $N = EM \in \mathbb{F}^{m' \times n}$ is $(r, tm' \cdot \frac{1}{10q})$-globally rigid.



ROW TO GLOBAL RIGIDITY

Theorem

Let $E \in \mathbb{F}^{m' \times m}$ be a q-query LDC. If $M \in \mathbb{F}^{m \times n}$ is (r, t)-rigid, then $N = EM \in \mathbb{F}^{m' \times n}$ is $(r, tm' \cdot \frac{1}{10q})$-globally rigid.



ROW TO GLOBAL RIGIDITY

Theorem

Let $E \in \mathbb{F}^{m' \times m}$ be a q-query LDC. If $M \in \mathbb{F}^{m \times n}$ is (r, t)-rigid, then $N = EM \in \mathbb{F}^{m' \times n}$ is $(r, tm' \cdot \frac{1}{10q})$-globally rigid.