Matrix Rigidity

Overview of Part I

Sasha Golovnev

September 21, 2020

Tools Used in Part I

Previous Lectures

Previous Lectures

$\Pr_{\text{Random obj}}[\text{obj is GOOD}] \Rightarrow \exists$ exists a GOOD obj.

- Probabilistic Method

Previous Lectures

Alg is often as good as randomness

- Probabilistic Method
- Algebraic Independence

Previous Lectures

- Probabilistic Method
- Algebraic Independence
- Polynomial Method

1. N objects => N polys.
2. These polys are lin. ind.
3. Dim of span of these polys is small => N is small

Previous Lectures

- Probabilistic Method
- Algebraic Independence
- Polynomial Method
- Zarankiewicz Problem

To upper bound # edges it often suffices to say graph doesn't contain small cliques

Previous Lectures

- Probabilistic Method
- Algebraic Independence
- Polynomial Method
- Zarankiewicz Problem
- Hölder's Inequality

From L_p to L_q

$\sum x_i^p \leq \quad \Rightarrow \quad \sum x_i^q \leq \ldots$

Sign-rank

Let $A \in \{1, -1\}^{n \times n}$. Then **signrk**$(A)$ is the minimum rank of B s.t. $sign(b_{ij}) = a_{ij}$ for all $i, i \in [n]$.

$B \in \mathbb{R}^{n \times n}$ $\quad$ $\text{sign}(b_{ij}) = a_{ij}$

$$\text{signrk}(A) = \min_{B} \text{rk}(B)$$

$$A = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$\text{rk}(A) = n-1$

$\text{signrk}(A) = 2$

Sign-rank

Let $A \in \{1, -1\}^{n \times n}$. Then **signrk**$(A)$ is the minimum rank of B s.t. $sign(b_{ij}) = a_{ij}$ for all $i, i \in [n]$.

Theorem (Paturi, Simon)

Unbounded-error communication complexity of a problem is $\log(\mathbf{signrk}(A))$ *of its communication matrix A.*

CC(F)

Alice & Bob — private Randomness
x y

compute $f(x,y)$ w.p. $> \frac{1}{2}$ (say, $\frac{1}{2} + 2^{-n}$)

Zero-patterns

n polys of t vars

- A vector $\sigma \in \{0,1\}^n$ is a zero-pattern of polynomials $p_1, \ldots, p_n$ of t variables if there exists $x \in \mathbb{R}^t$ s.t. $p_i(x) = 0$ iff $\sigma_i = 0$

$n=2$ $t=1$

$p_1 = x-3$

$p_2 = x^2+9$

Ex. $\sigma_1 = (0,1)$ is a zero-pattern

$x=3 \Rightarrow p_1 = 0$, $p_2 \neq 0$ — $(0,1)$ a zero-pattern

Ex. $\sigma_2 = (1,0)$ is a z. patt?

No, it is not.

Zero-patterns

- A vector $\sigma \in \{0,1\}^n$ is a zero-pattern of polynomials $p_1, \ldots, p_n$ of t variables if there exists $x \in \mathbb{R}^t$ s.t. $p_i(x) = 0$ iff $\sigma_i = 0$
- The number of zero-patterns is $\leq 2^n$

Zero-patterns

- A vector $\sigma \in \{0,1\}^n$ is a zero-pattern of polynomials $p_1, \ldots, p_n$ of t variables if there exists $x \in \mathbb{R}^t$ s.t. $p_i(x) = 0$ iff $\sigma_i = 0$
- The number of zero-patterns is $\leq 2^n$

Theorem

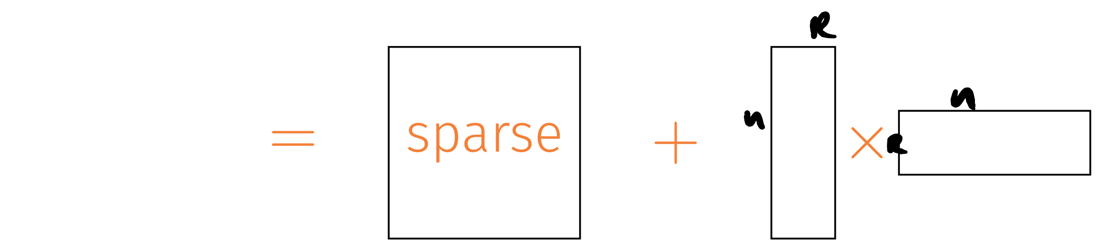
For constant degree polynomials, the number of zero-patterns is $\lesssim \binom{n}{t} \ll 2^n$ (for small t)

Zero-patterns

- Non-rigid M:

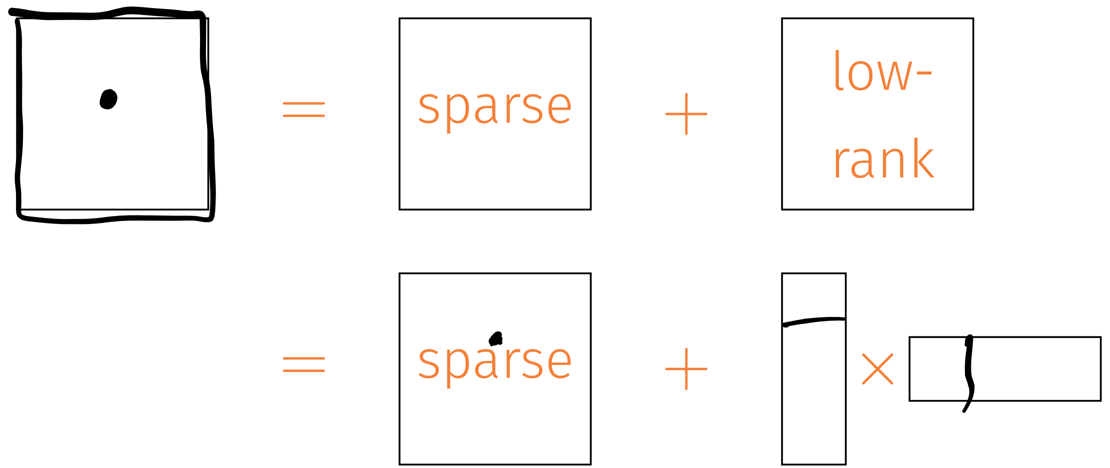
Zero-patterns

- Non-rigid M:



Zero-patterns

- Non-rigid M:



- Each (out of n^2) entries on the left is a degree-2 polynomial of the entries on the right

Zero-patterns

- By the zero-pattern theorem, there are only a few zero-non-zero matrices low-degree polynomials can generate

Zero-patterns

- By the zero-pattern theorem, there are only a few zero-non-zero matrices low-degree polynomials can generate
- In particular, only a few $\{0, 1\}$-matrices

Zero-patterns

- By the zero-pattern theorem, there are only a few zero-non-zero matrices low-degree polynomials can generate
- In particular, only a few $\{0, 1\}$-matrices
- Therefore, a random $\{0, 1\}$-matrix is rigid

Sign-patterns. Example

Theorem

The number of sign*-patterns of n constant-degree polynomials of t variables is (also)* $\lesssim \binom{n}{t}$.

$\sigma \in \{+1, -1\}^n$ is sign-pattern if $\exists x$

$p_i(x) > 0$ if $\sigma_i = +1$

$p_i(x) < 0$ if $\sigma_i = -1$

Ex.

$p_1 = x - 3$

$p_2 = x^2 - 1$

$\sigma_1 = (+1, +1)$ is a sign p.

$x = 10 \Rightarrow p_1 > 0 \;\; p_2 > 0$

$\sigma_2 = (+1, -1)$

is not a sign pattern

$x^2 - 1 < 0 \Leftrightarrow |x| < 1$

$p_1 = x - 3 < 0$

Sign-patterns. Example

Theorem

The number of sign-patterns of n constant-degree polynomials of t variables is (also) $\lesssim \binom{n}{t}$.

Theorem

There exist matrices of high sign-rank. (There exist problems of high unbounded-error communication complexity.)

Theorem

There exist matrices of high sign-rank. (There exist problems of high unbounded-error communication complexity.)

Theorem

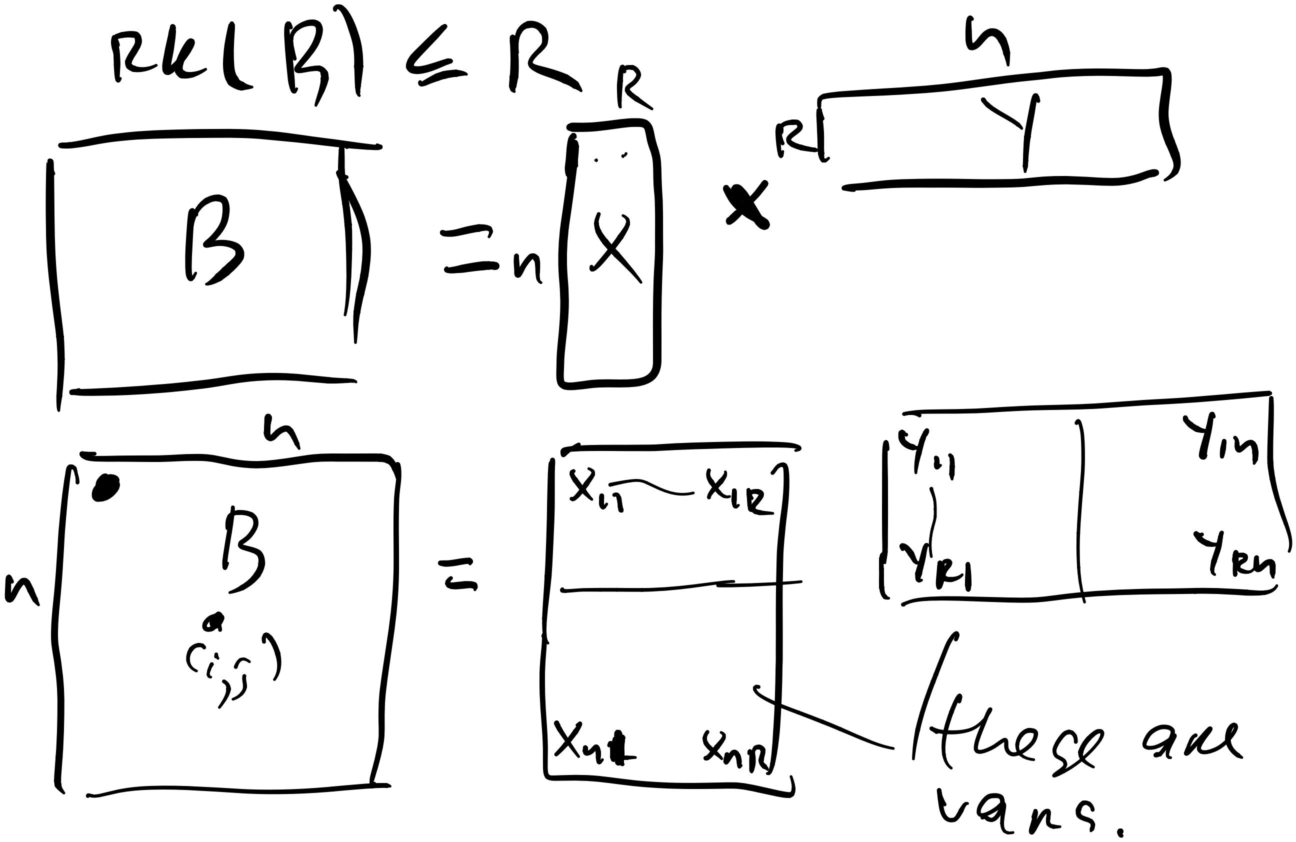
The number of sign-patterns of n constant-degree polynomials of t variables is (also) $\lesssim \binom{n}{t}$.

Pf. $A \in \{\pm 1\}^{n \times n}$

$\text{signrank}(A) \le R$

$\exists B \in \mathbb{R}^{n \times n}$ $\text{sign}(b_{ij}) = a_{ij}$

$\text{rk}(B) \le R$



$2nR$

n^2 polys.

each poly is of deg 2.

IF signrk$(A) \leq R$

$\exists$ $x_{11} \ldots x_{nR}, y_{11} \ldots y_{Rn}$

s.t. $rk(B) \leq R$

fixed set of n^2 polys.

$\binom{n^2}{\#vars}$ diff $\{\pm 1\}^{n \times n}$ matrices

\# As of low signrk

$$\leq \binom{n^2}{\#vars} = \binom{n^2}{2nR}$$

$R = \varepsilon n$

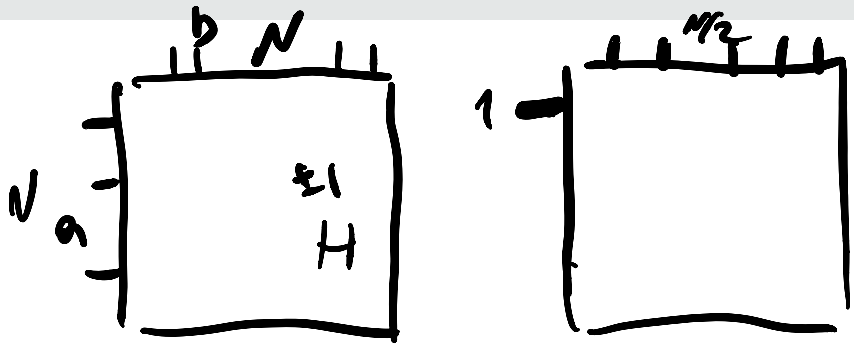
$$= \binom{n^2}{2\varepsilon n^2} \leq 2^{n^2 H(2\varepsilon)} =$$

$$= 2^{n^2 \cdot \delta} << 2^{n^2} - \text{\# of matrice}$$

Walsh-Hadamard Matrix. Example

Lemma (Lindsey's Lemma)

For any submatrix $H' \in \mathbb{C}^{a \times b}$ of Hadamard $H \in \mathbb{C}^{N \times N}$, the absolute value of the sum of all entries in H' is at most $\sqrt{abN}$.



$ab \geq \Omega(N)$

Hadamard Matrix. Example

Lemma (Lindsey's Lemma)

For any submatrix $H' \in \mathbb{C}^{a \times b}$ of Hadamard $H \in \mathbb{C}^{N \times N}$, the absolute value of the sum of all entries in H' is at most $\sqrt{abN}$.

$ab \geq 10000N$ $\sqrt{abN} = 100N$

- Hadamard is quasirandom: for $ab \geq \Omega(N)$, every $a \times b$ submatrix has 49% − 51% of 1 and −1.

Expl. matrix has some prop of random matrix

Hadamard Matrix. Example

Lemma (Lindsey's Lemma)

For any submatrix $H' \in \mathbb{C}^{a\times b}$ of Hadamard $H \in \mathbb{C}^{N\times N}$, the absolute value of the sum of all entries in H' is at most $\sqrt{abN}$.

- Hadamard is quasirandom: for $ab \geq \Omega(N)$, every $a \times b$ submatrix has $49\% - 51\%$ of 1 and -1.
- Implies circuit lower bounds, extractors, Ramsey graphs, discrepancy, . . .

Spectral Methods

- We used relations between matrix norms to prove that for any $H' \in \mathbb{R}^{a \times b}$ submatrix of Hadamard H,

$$\mathbf{rank}(H') \geq ab/N\,.$$

Spectral Methods

- We used relations between matrix norms to prove that for any $H' \in \mathbb{R}^{a \times b}$ submatrix of Hadamard H,

$$\text{rank}(H') \geq ab/N\,.$$

- After a few changes in Hadamard, there remains an untouched submatrix of size $a \times b$

Spectral Methods

- We used relations between matrix norms to prove that for any $H' \in \mathbb{R}^{a \times b}$ submatrix of Hadamard H,

$$\mathrm{rank}(H') \geq ab/N\,.$$

- After a few changes in Hadamrd, there remains an untouched submatrix of size $a \times b$
- The rank of this matrix is large, which implies rigidity of Hadamard

Spectral Methods. Example

Theorem (Forster's Theorem)

For every $A \in \{-1, 1\}^{m \times n}$,

$$\operatorname{signrk}(A) \geq \frac{\sqrt{mn}}{\|A\|_2}.$$

$H_N \in \{\pm 1\}^{N \times N}$

$\operatorname{signrk}(H) \geq \frac{N}{\|H\|_2} = \frac{N}{\lambda_1(H)} = \sqrt{N}$

$N = 2^n$

$\operatorname{signrk} \geq \sqrt{N} = 2^{n/2}$

$CC = \log(\operatorname{signrk}) = n/2$

Spectral Methods. Example

Theorem (Forster's Theorem)

For every $A \in \{-1,1\}^{m\times n}$,

$$\operatorname{signrk}(A) \geq \frac{\sqrt{mn}}{\|A\|_2}\,.$$

Recall that $\|H_N\|_2 = \sqrt{N}$.

Spectral Methods. Example

Theorem (Forster's Theorem)

For every $A \in \{-1,1\}^{m\times n}$,

$$\mathrm{signrk}(A) \geq \frac{\sqrt{mn}}{\|A\|_2}\,.$$

Recall that $\|H_N\|_2 = \sqrt{N}$.

Corollary

$$\mathrm{signrk}(H_N) \geq \sqrt{N}\,.$$

Error-Correcting Codes

A linear code C is a subspace of $\mathbb{F}^n$ where every non-zero vector c has

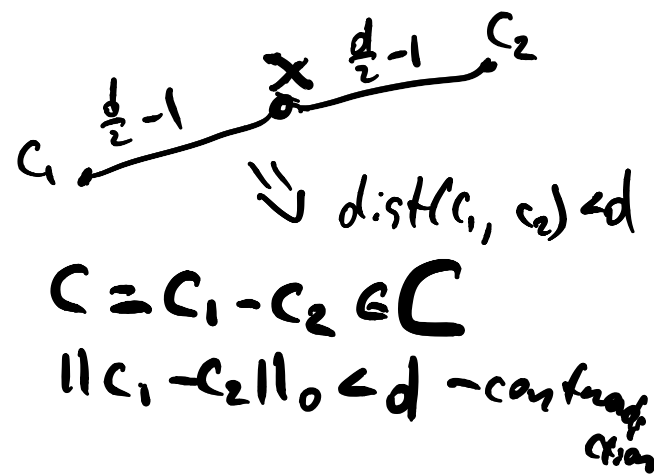
$$\|c\|_0 \geq d\,.$$

Error-Correcting Codes

A linear code C is a subspace of $\mathbb{F}^n$ where every non-zero vector c has

$$\|c\|_0 \geq d\,.$$

In particular, for every $x \in \mathbb{F}^n$ there is at most one $c \in C$ s.t. $\|x - c\|_0 < d/2$.



Error-Correcting Codes

A linear code C is a subspace of $\mathbb{F}^n$ where every non-zero vector c has

$$\|c\|_0 \geq d\,.$$

In particular, for every $x \in \mathbb{F}^n$ there is at most one $c \in C$ s.t. $\|x - c\|_0 < d/2$.

Proposition

For any finite field $\mathbb{F}$, there exists an explicit family of linear error correcting codes over $\mathbb{F}$ of dimension $k = n/4$ and minimum distance $d = \delta n$ for a constant $\delta > 0$.

Static Data Structures. Examples

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

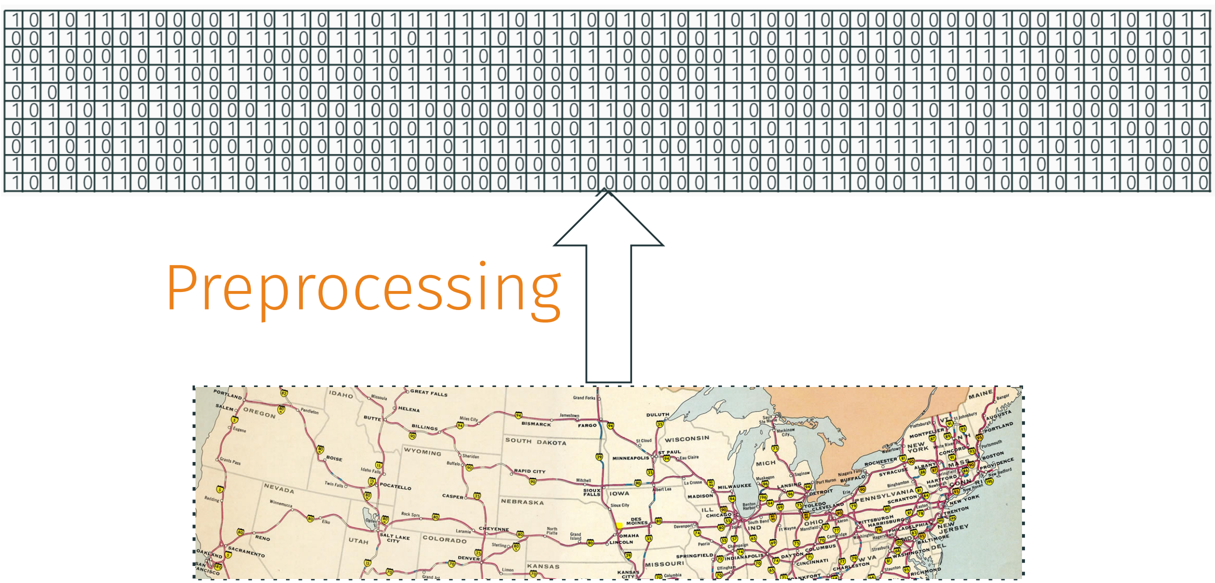
Static Data Structures. Examples

- 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



Static Data Structures

Queries

Preprocessing

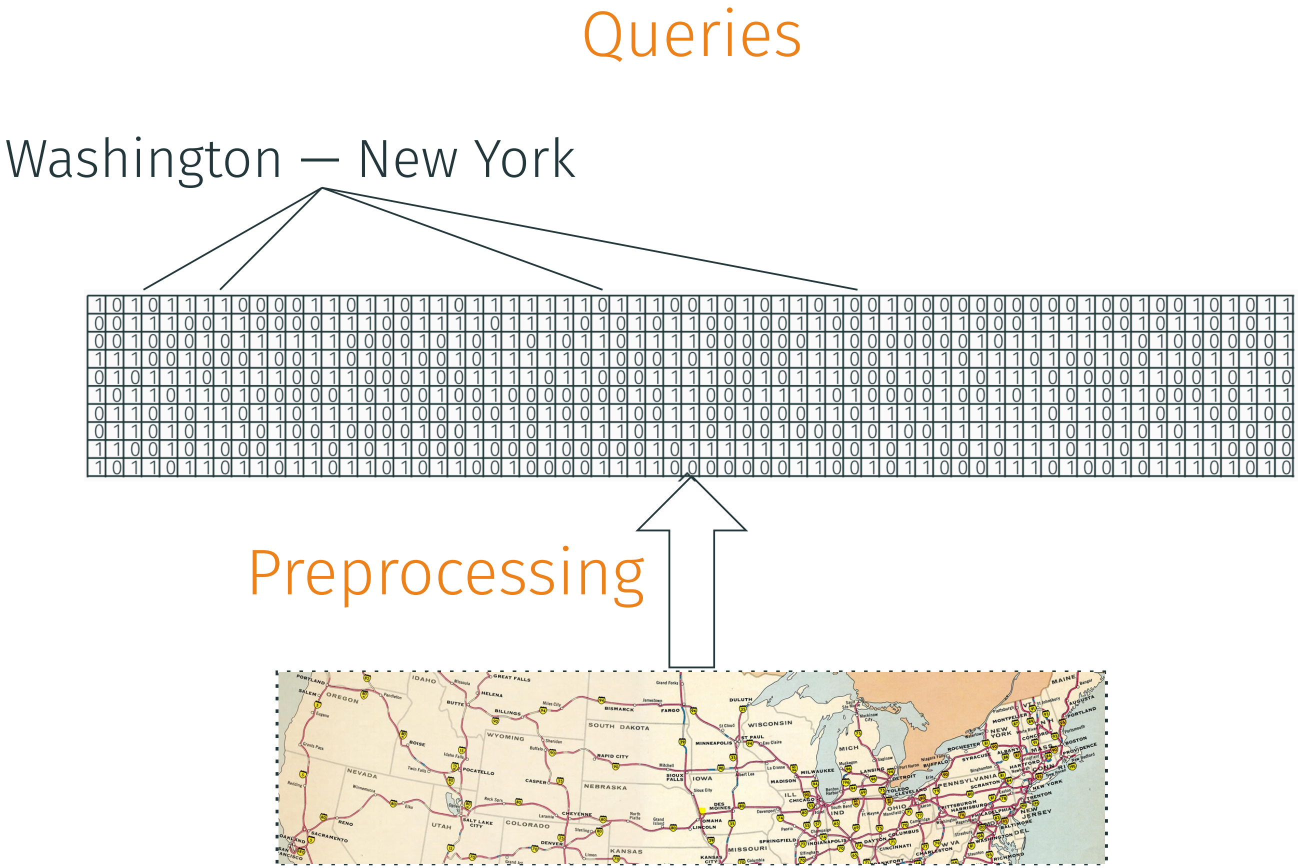
Static Data Structures

Queries

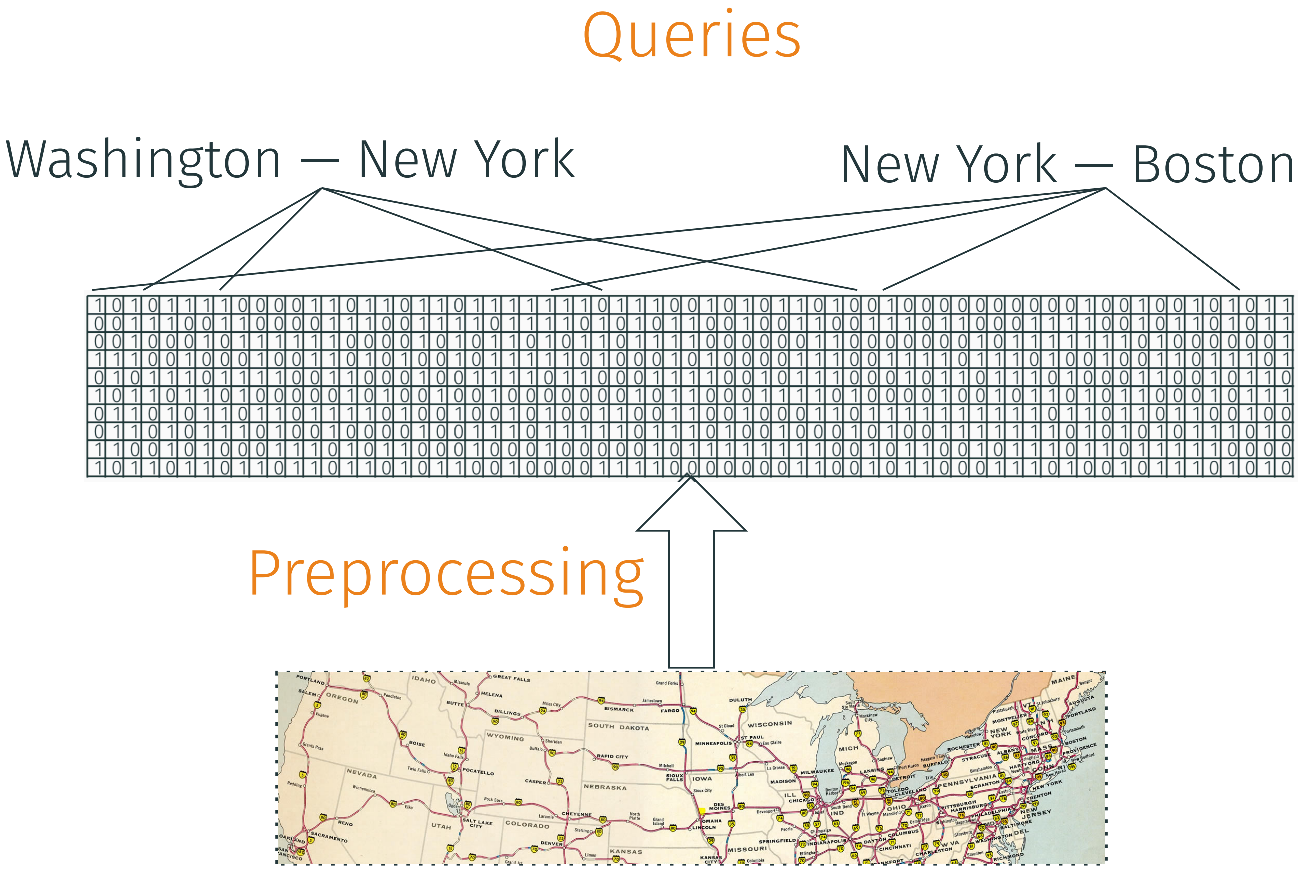
Washington — New York

Preprocessing

Static Data Structures



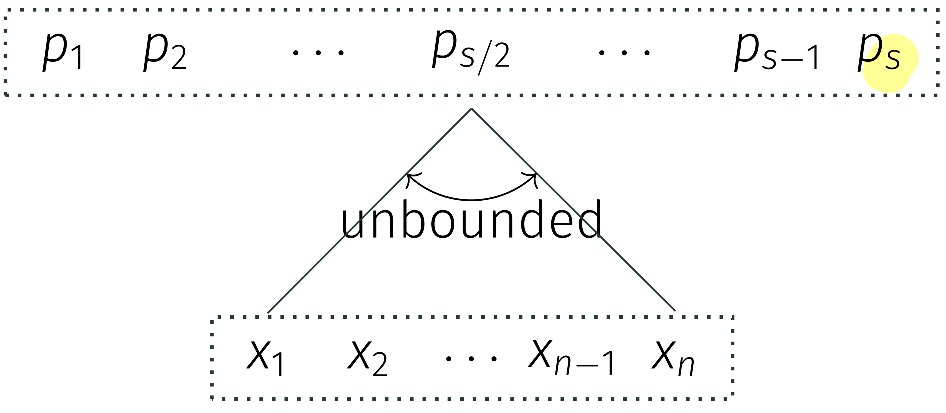
Static Data Structures



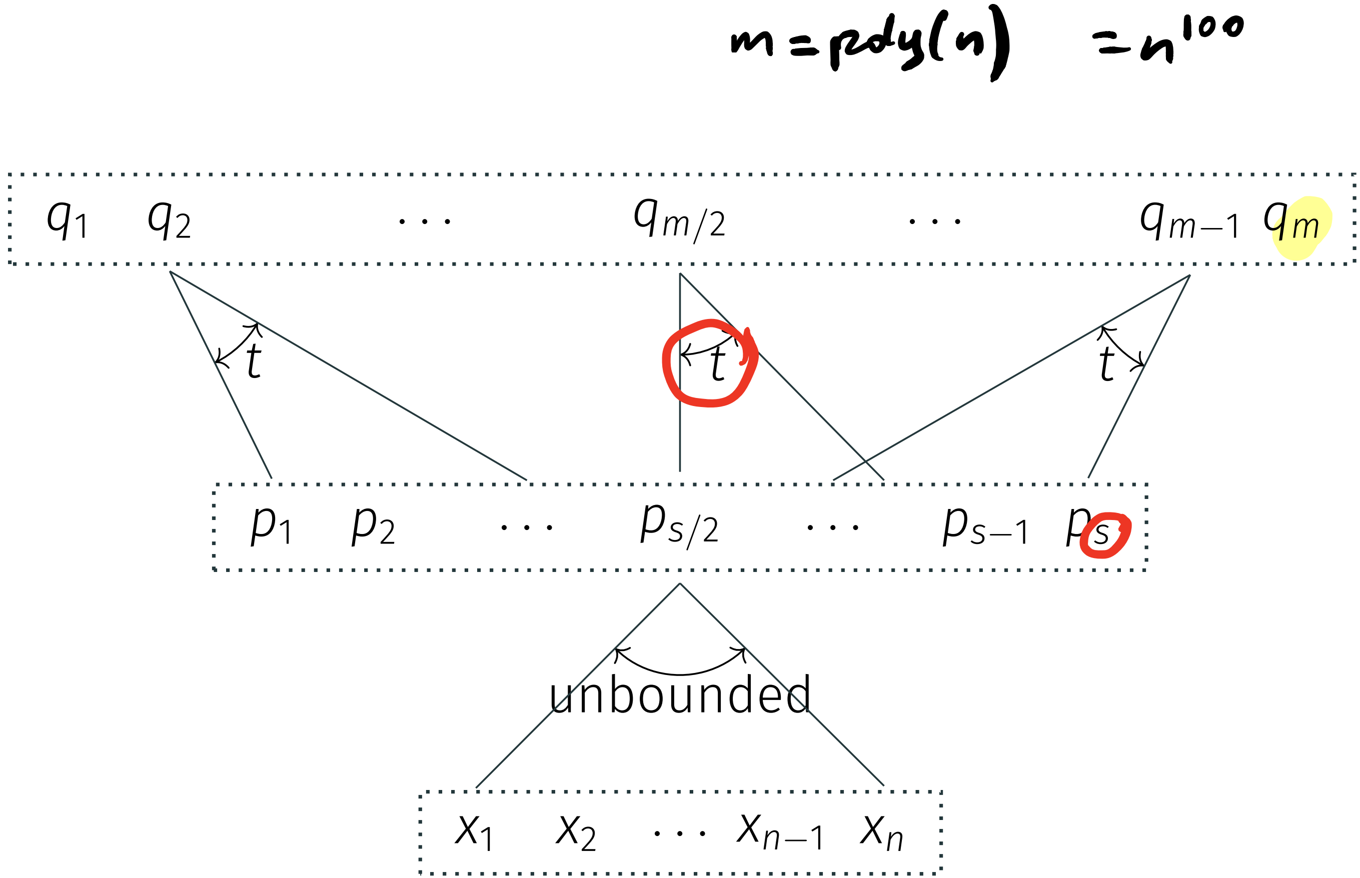
Static Data Structures. Definition

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

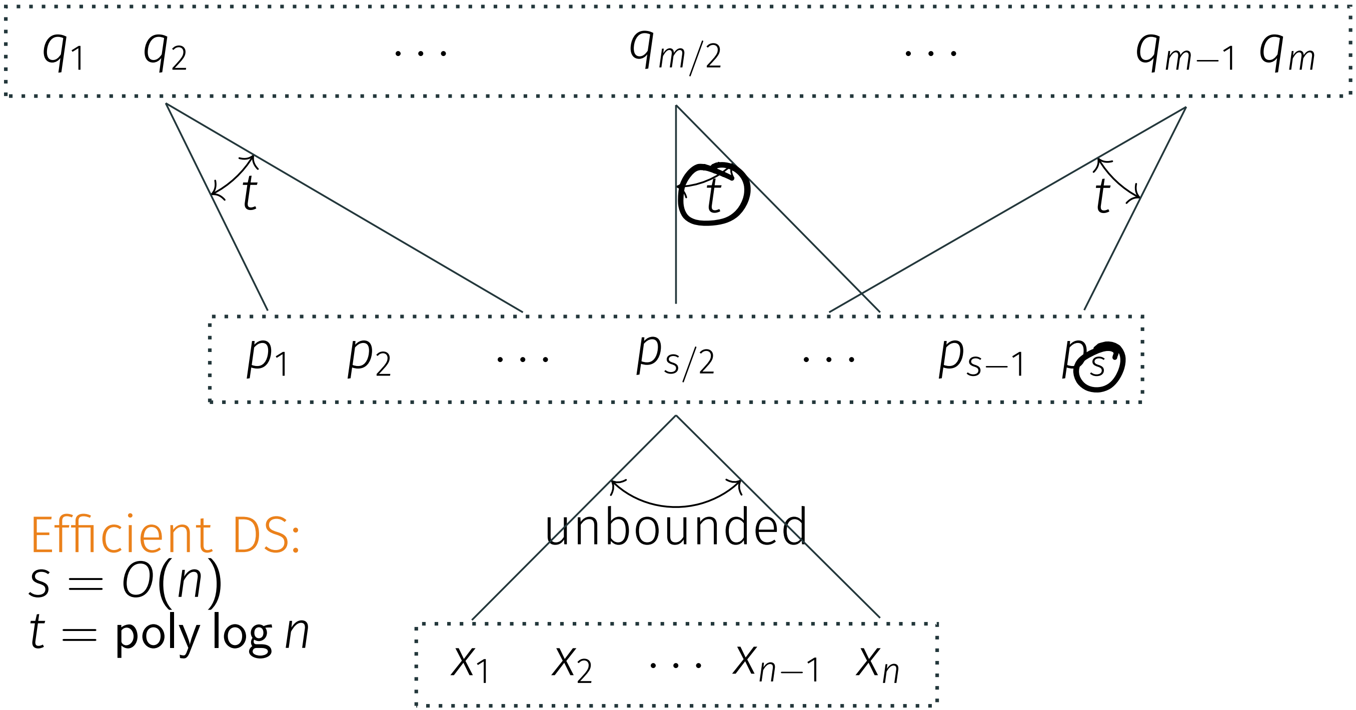
Static Data Structures. Definition



Static Data Structures. Definition



Static Data Structures. Definition



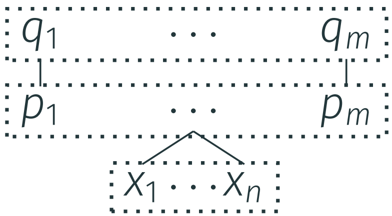
Efficient DS:
$s = O(n)$
$t = \text{poly} \log n$

DS Lower Bounds

- Two trivial solutions:

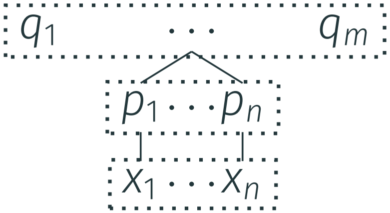
DS Lower Bounds

- Two trivial solutions:
  - $s = m, t = 1$



DS Lower Bounds

- Two trivial solutions:
  - $s = m, t = 1$
  - $s = n, t = n$



DS Lower Bounds

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

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 ≈ 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]:

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

- $s = O(n) \implies t \geq \Omega(\log n)$

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]:

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

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

DS Lower Bound. Proof

Theorem

Let $f\colon \mathbb{F}^n \to \mathbb{F}^m$ be a good error correcting code. Then for any data structure computing f, we have

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

Theorem

Let $f\colon \mathbb{F}^n \to \mathbb{F}^m$ be a good error correcting code. Then for any data structure computing f, we have

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

code maps

$GF^n \to F^m$

$x = x_1 \dots x_n \in F$

$\Downarrow$

$y = y_1 \dots y_m \in F$

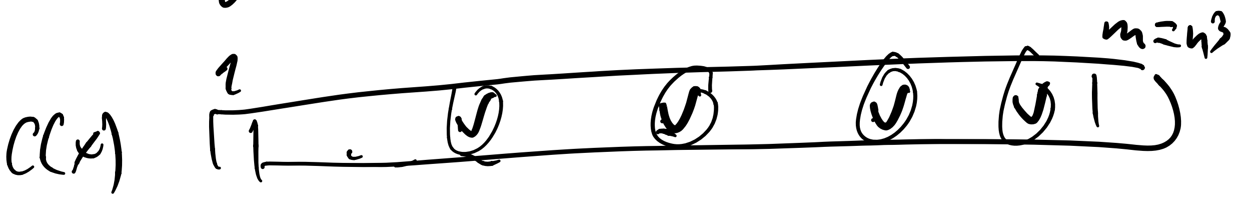
DS problem:

$x \to C(x)$

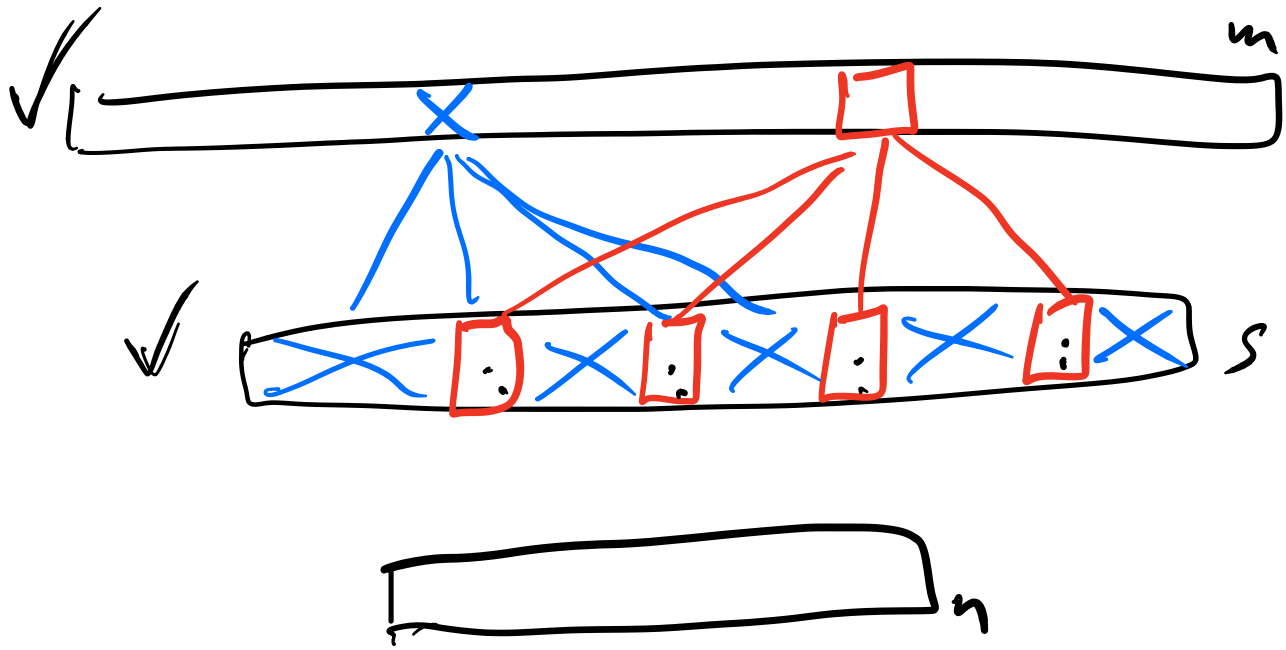
$m = n^3$

Good code: given 10n outputs (out of m), I can find decode x.

$x \in F^n \Rightarrow C(x) \in F^m$



10n values



Compute 10n outputs ⇒ compute input

Cell sampling (w. prob. p)

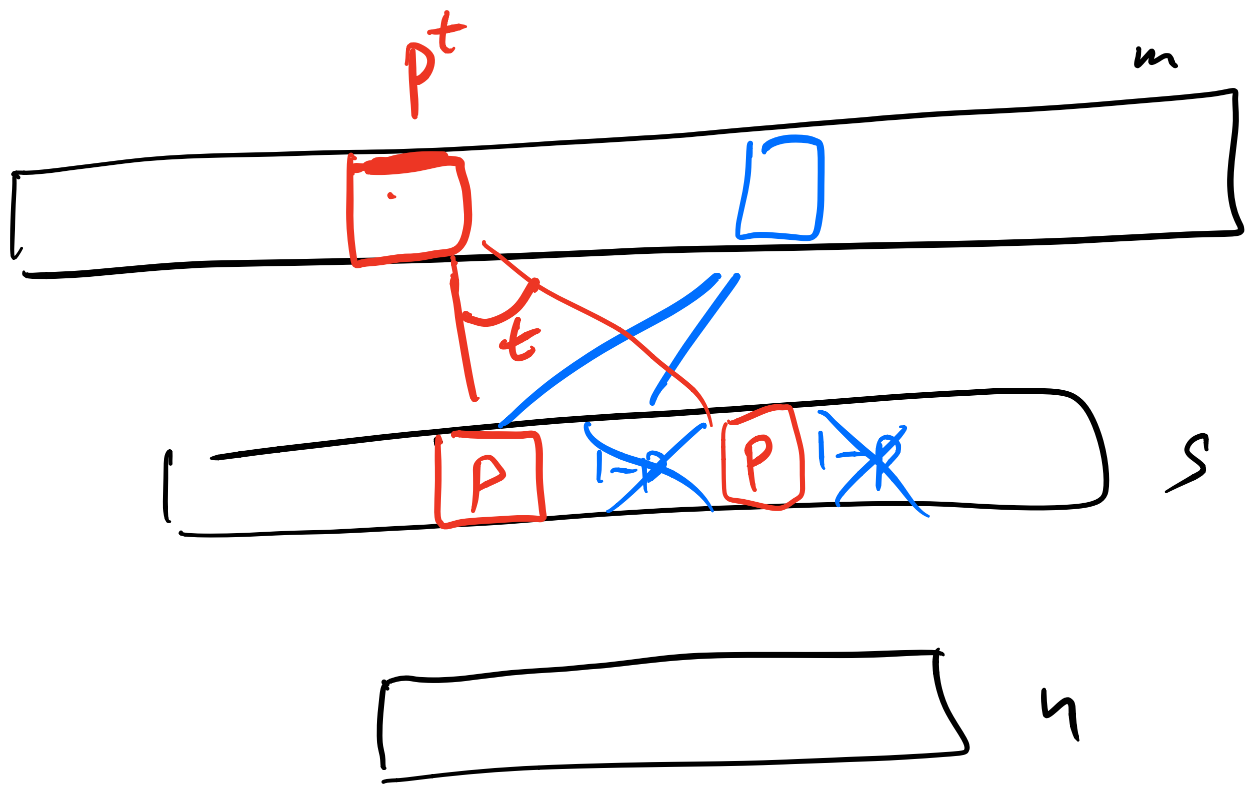
Plan: delete data

memory cells < n

outputs > 10n

10n outputs ⇒ compute input

< n memory cells encode inpu-



$p \cdot s < n \qquad p < \frac{n}{s}$

$m \cdot p^t$ outputs

$m \cdot \left(\frac{n}{s}\right)^t \geq 10n$ outputs

$\left(\frac{n}{s}\right)^t \geq \frac{10n}{m}$

$t \leq \frac{\log\left(\frac{m}{n}\right)}{\log\left(\frac{s}{n}\right)}$

If $t < \frac{\log(\frac{m}{n})}{\log(\frac{s}{n})}$, then

there exist $n-1$ memory cells from which I always compute $\geq 10n$ outputs.

From the code property $\Rightarrow$

From any $10n$ outputs I can recover all n cells of input

Contradiction: $n-1$ cells encode n cells.

Thus, $t \geq \frac{\log(\frac{m}{n})}{\log(s/n)}$