APPLICATIONS

- Streaming
- Property Testing
- game theory
- TIME/SPACE Turing Machine LBs

TODAY

- Circuit complexity
- Proof complexity
- Extension Complexity
- clique/coclique, graph Theory, Learning Partial Functions

Main CC Lower Bounds

UDISJ : disjointness with promise that either $|x \cap y| = 0$ or $|x \cap y| = 1$

Theorem

$$BPP^{cc}(DISJ) = \Omega(n)$$
$$BPP^{cc}(UDISJ) = \Omega(n)$$
$$coNP^{cc}(UDISJ) = \Omega(n)$$

Theorem

The k-player NOF randomized cc of DISJ, UDISJ is $\Omega\left(\frac{n}{2^k}\right)$

We will prove these in a couple of weeks

APPLICATIONS

- Streaming
- Property Testing
- game theory
- TIME/SPACE Turing Machine LBs
- Circuit complexity
- Proof complexity
- Extension Complexity
- clique/coclique, graph Theory, Learning Partial Functions

STREAMING LOWER BOUNDS

$S \in [n]^m$ is a length m stream

Computing frequency moments of S:

Let $M_i = |\{j \in [m] \mid S_j = i\}|$

The k^{th} frequency moment of S, $F_k = \sum_{i=1}^{n} M_i^k$

F_0 = # distinct elements in stream

F_1 = length of stream

F_∞ = # occurences of most frequent item

STREAMING LOWER BOUNDS

$S \in [n]^m$ is a length m stream

computing frequency moments of S:

Let $M_i = |\{j \in [m] \mid S_j = i\}|$

The k^{th} frequency moment of S, $F_k = \sum_{i=1}^{n} M_i^k$

F_0 = # distinct elements in stream

F_1 = length of stream

F_∞ = # occurrences of most frequent item

Theorem F_0, F_2 can be approx'd to within a $(1 \pm \varepsilon)$ factor (wp $\geq 1-\delta$)

in space $O\left(\frac{(\log n + \log m) \log \frac{1}{\delta}}{\varepsilon^2}\right)$

STREAMING LOWER BOUNDS

2	10	14	1	1	3	3	10	7	5	....

$S \in [n]^m$ is a length m stream

computing frequency moments of S:

$$\left(\sum M_i^k\right)^{\frac{1}{k}}$$

Let $M_i = |\{j \in [m] \mid S_j = i\}|$

The k^{th} frequency moment of S, $F_k = \sum_{i=1}^{n} M_i^k$

F_0 = # distinct elements in stream

F_1 = length of stream

F_∞ = # occurences of most frequent item

Theorem computing F_∞ requires $\Omega(\min\{m,n\})$ space

stronger: any randomized alg for F_∞ to within $(1 \pm .2)$ factor wp $\geq \frac{2}{3}$ requires space $\Omega(\min\{m,n\})$.

Theorem computing F_∞ requires $\Omega(n)$ space/memory $(m=n)$

Pf Reduction from DISJ $\longrightarrow$ low-space streaming alg for F_∞

Let A be space c streaming alg

Alice: $x \longrightarrow$ stream $a_x = \{i \mid x_i = 1\}$ $\quad 011011 \longrightarrow 2, 3, 5, 6$

Bob: $y \longrightarrow$ stream $b_x = \{j \mid y_j = 1\}$ $\quad 100100 \longrightarrow 1, 4$

Fact $DISJ(x,y) = 1 \implies F_\infty(a_x b_x) = 2$

$DISJ(x,y) = 0 \implies F_\infty(a_x b_x) = 1$

Theorem computing F_∞ requires $\Omega(n)$ space/memory

Pf Reduction from DISJ $\longrightarrow$ low-space streaming alg for F_∞

Let A be space c streaming alg

Alice: $x \longrightarrow$ stream $a_x = \{i \mid x_i = 1\}$ 011011 $\longrightarrow$ 2, 3, 5, 6

Bob: $y \longrightarrow$ stream $b_x = \{j \mid y_j = 1\}$ 100100 $\longrightarrow$ 1, 4

Fact DISJ$(x, y) = 1 \implies F_\infty(a_x\, b_x) = 2$
DISJ$(x, y) = 0 \implies F_\infty(a_x b_x) = 1$

Simulation Alice simulates A on a_x & sends content of memory (c bits) to Bob; then Bob simulates rest of computation on b_y

MORE STREAMING LOWER BOUNDS

Previous LB actually showed something stronger:

Thm Any randomized streaming alg that for any stream S of length m computes F_∞ to within $(1 \pm .2)$ factor (with prob $> 2/3$) requires space $\Omega(\min\{m, n\})$.

Thm For $k \neq 1$ every randomized streaming alg for computing F_k exactly requires space $\Omega(\min\{m, n\})$

In our reduction F_∞ is 1 vs 2

so a factor of 2 difference.

For $k \neq 1$ the correct value will still be different in the 2 cases

MORE STREAMING LOWER BOUNDS

In contrast, we have very low space approx. algs for F_0 and F_2

Thm F_0, F_2 can be approx'd to within $(1 \pm \varepsilon)$ factor with prob $\geq (1-\delta)$ using space $O(\varepsilon^{-2}(\log n + \log m)\log\frac{1}{\delta})$

APPLICATIONS

- Streaming
- Property Testing
- game theory
- TIME/SPACE Turing Machine LBs
- Circuit complexity
- Proof complexity
- Extension Complexity
- Data structures

PROPERTY TESTING

Let D = domain (usually $D = \{0,1\}^n$)
R = range

A property P is a set of functions from $D \to R$

Examples

① Linearity. $D = \mathbb{F}^n$ $\mathbb{F}_2$
$R = \mathbb{F}$

P = set of all linear functions

② Monotonicity $D = \{0,1\}^n$, $R = \{0,1\}$
P = set of all monotone functions $f : \{0,1\}^n \to \{0,1\}$

③ graph testing $D = \{0,1\}^{\binom{n}{2}}$ $R = \{0,1\}$
P = all graphs that have a k-clique, etc...

<u>goal of property testing</u>: given a very large input (like a graph or Boolean function on n inputs) want to look at <u>few</u> places in input to decide if it is close to an input with property P or <u>far</u> from all inputs in P

View function (input) as a vector indexed by D

<u>Defn</u> f is ε-far from P if $\forall g \in P$, f and g differ in $\geq \varepsilon |D|$ entries.

ie. changing f to some $g \in P$ requires changing at least an ε fraction of its values given property $\mathcal{P}$ defined wrt D, R

on input $f: D \to R$, determine if

① $f \in \mathcal{P}$ or

② f is ε-far from $\mathcal{P}$

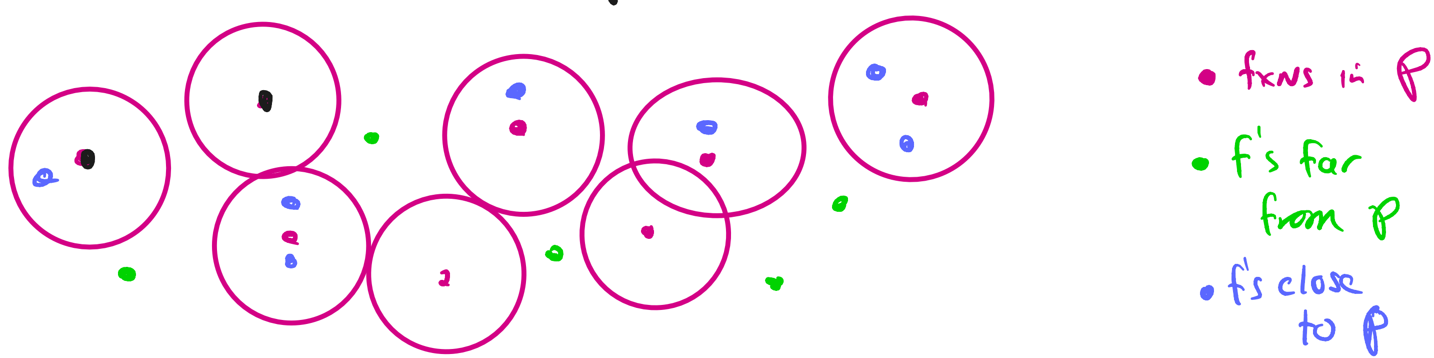
A <u>tester</u> for $\mathcal{P}$ queries input f (a decision tree)

Query complexity of $\mathcal{P}$ =

min. decision tree depth over all testers for $\mathcal{P}$

$\varepsilon \sim \frac{1}{10}$

$\frac{1}{100}$



Example 1 : Linearity Testing (over $\mathbb{F}_2$)

Input $f : \{0,1\}^n \to \{0,1\}$ (f as a vector of length 2^n)

Is f ε-close to a parity function?

Parity fxns = Linear fxns $\equiv f(x \oplus y) = f(x) \oplus f(y) \quad \forall x, y \in \{0,1\}^n$

BLR Test:

Repeat $\Theta(\frac{1}{\varepsilon})$ times

- Pick $x, y \sim \{0,1\}^n$ unif. at random
- If $f(x \oplus y) \neq f(x) \oplus f(y)$ halt & reject

If havent yet rejected then ACCEPT

Theorem

With constant probability, every function ε-far from linear is rejected

Example 2: Monotone graph Properties

Boolean case: $f: \{0,1\}^n \to \{0,1\}$.

Picture f as choosing a subset of vertices of n-dim boolean hypercube

Let (b, x_{-i}) be an assignment where i^{th} bit is b, remaining $n-1$ bits are x_{-i}

then f is monotone if $\forall i \in [n]\ \forall x_{-i}\quad f(0, x_{-i}) \leq f(1, x_{-i})$

Monotonicity Test

Repeat $O(\frac{n}{\varepsilon})$ times:

- Pick i, x_{-i} at random.
- If $f(0, x_{-i}) > f(1, x_{-i})$ halt + reject

If havent rejected yet, ACCEPT

Thm With prob $> \frac{2}{3}$ every function ε-far from monotone is rejected

Example 2: Monotone graph Properties

	UB		LB	
Boolean	[GGLRS 2000]	$O(\frac{n}{\varepsilon})$	[FLNRRS 2002]	$\Omega(\sqrt{n})$ nonadaptive $\Omega(\log n)$ adaptive
	[KMS 2015]	$O(\frac{\sqrt{n}}{\varepsilon^2})$	[BB '18]	$\Omega(n^{1/3})$ adaptive
Range R	[GGLRS 2000]	$O(n\,\lvert R\rvert/\varepsilon)$	[BBM 2012]	$\Omega(n)$, $\lvert R\rvert = \Omega(\sqrt{n})$ $\Omega(\lvert R\rvert^2)$
	[DGLRRS '99]	$O(\frac{n}{\varepsilon}\log \lvert R\rvert)$		

NEXT

LBs: many excellent LBs by
Chen, Servedio, Tan, Waingarten, Xie

MONOTONICITY TESTING LOWER BOUNDS

General Template:

- Map 1-inputs (x,y) of a hard cc problem (UDISJ) to functions $h_{x,y} \in \mathcal{P}$
- Map 0-inputs (x,y) to $h_{x,y}$ that are far from $\mathcal{P}$
- Use efficient tester Π for $\mathcal{P}$, plus short protocol to evaluate $h_{x,y}$ to solve UDISJ

MONOTONICITY TESTING LOWER BOUNDS

Lemma For $A, B \subseteq [n]$, let $h_{A,B} : \{0,1\}^n \to \mathbb{Z}$ by

$$h_{A,B}(x) = 2|x| + (-1)^{|x \cap A|} + (-1)^{|x \cap B|}$$

Then (i) If $A \cap B = \phi \longrightarrow h_{A,B}$ is monotone

(ii) If $|A \cap B| = 1 \longrightarrow h_{A,B}$ is ε-far from monotone

MONOTONICITY TESTING LOWER BOUNDS

Lemma For $A, B \subseteq [n]$, let $h_{A,B} : \{0,1\}^n \to \mathbb{Z}$ by

$$h_{A,B}(x) = 2|x| + (-1)^{|x \cap A|} + (-1)^{|x \cap B|}$$

Then (i) If $A \cap B = \phi \longrightarrow h_{A,B}$ is monotone

(ii) If $|A \cap B| = 1 \longrightarrow h_{A,B}$ is ε-far from monotone

Assuming lemma, let Q be a monotonicity tester $(\varepsilon = 1/8)$ given input (A,B) to UDISJ Alice/Bob simulate Q on $h_{A,B}$:

Let $x \subseteq [n]$ be next query Q asks $(h_{A,B}(x)?)$

Alice sends $(-1)^{|x \cap A|}$

Bob sends $(-1)^{|x \cap B|}$

Cost per query = 2.

$\therefore$ monotonicity testing $(\varepsilon = 1/8)$ requires $\Omega(n)$ queries

MONOTONICITY TESTING LOWER BOUNDS

Lemma For $A, B \subseteq [n]$, Let $h_{A,B} : \{0,1\}^n \to \mathbb{Z}$ by

$$h_{A,B}(x) = 2|x| + (-1)^{|x \cap A|} + (-1)^{|x \cap B|}$$

Then (i) If $A \cap B = \phi \longrightarrow h_{A,B}$ is monotone

(ii) If $|A \cap B| = 1 \longrightarrow h_{A,B}$ is ε-far from monotone

Proof

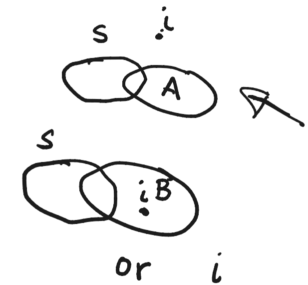
(1) Want to show: A, B disjoint $\Rightarrow \forall S, i \notin S \quad h_{AB}(S \cup i) - h_{AB}(S) \geq 0$

Since A, B disjoint, either $i \notin A$ or $i \notin B$. Assume wlog $i \notin A$. Then

$$h_{AB}(S \cup i) - h_{AB}(S) = 2 + \boxed{0} + (-1)^{|S \cap B| + 1} - (-1)^{|S \cap B|}$$

$$\geq 2 + 0 - 2$$

$$\geq 0$$



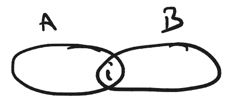
MONOTONICITY TESTING LOWER BOUNDS

Lemma For $A, B \subseteq [n]$, let $h_{A,B} : \{0,1\}^n \to \mathbb{Z}$ by

$$h_{A,B}(x) = 2|x| + (-1)^{|x \cap A|} + (-1)^{|x \cap B|}$$

Then (i) If $A \cap B = \phi \longrightarrow h_{A,B}$ is monotone

(ii) If $|A \cap B| = 1 \longrightarrow h_{A,B}$ is ε-far from monotone



Proof Let $A \cap B = i$

Claim: $Pr[(|S \cap A| \text{ is even}) \text{ and } (|S \cap B| \text{ is even})] \geq \frac{1}{4}$

When $|S \cap A|$ and $|S \cap B|$ are both even

$$h_{AB}(S \cup i) - h_{AB}(S) = 2|S| + 2 - 2|S| + (-1) - (1) + (-1) - (1) = -2$$

so for at least $\frac{1}{4} 2^{n-1} = \frac{1}{8} 2^n$ choices of S, $h_{AB}(S \cup i) < h_{AB}(S)$

so h_{AB} is $\frac{1}{8}$-far from monotone.

MONOTONICITY TESTING LOWER BOUNDS

The lower bound is $\Omega(n)$ as long as $|R| \geq n$.

This can be improved to show same LB $\Omega(n)$ for $|R| \geq \sqrt{n}$

More generally can prove $\Omega(|R|^2)$ LB.

OPEN

For testing monotonicity of Boolean functions best LB is $\tilde{\Omega}(n^{1/3})$, whereas best UB is $O(\sqrt{n})$

APPLICATIONS

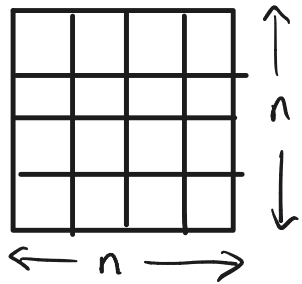
- Streaming
- Property Testing
- game theory (Pure + Approximate Nash)
- TIME/SPACE Turing Machine LBs
- Circuit complexity
- Proof complexity
- Extension Complexity
- Data structures

GAME THEORY: PURE NASH EQUILIBRIUM

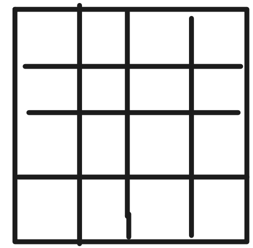
<u>NASH</u>:

Two players I, II have payoff matrices (zero sum)

I



II



A <u>pure Nash equilibrium</u> is a pair (i^*, j^*) st

i^* strategy is optimal if Bob plays j^* +

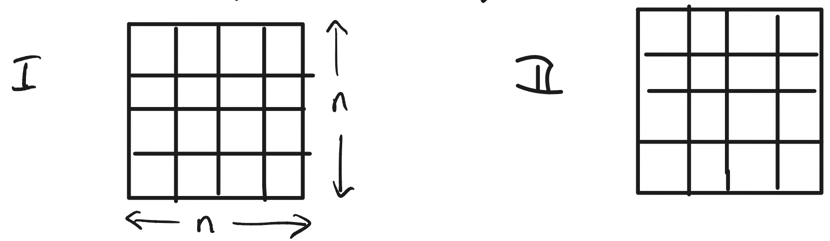
similarly j^* is optimal if Alice plays i^*

<u>Lemma</u> computing whether a pure Nash equilibrium exists requires $\Omega(n^2)$ cc

PURE NASH EQUILIBRIUM

NASH:

Two players I, II have payoff matrizes (zero sum)



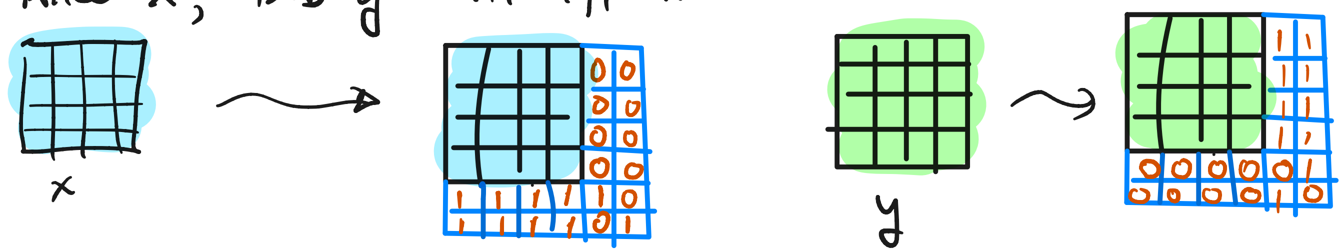
A pure Nash equilibrium is a pair (i^*, j^*) st

i^* strategy is optimal if Bob plays j^* +

similarly j^* is optimal if Alice plays i^*

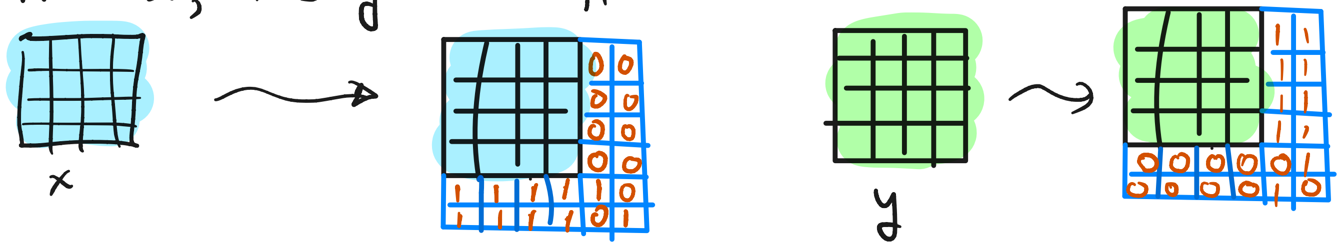
Lemma computing whether a pure Nash equilibrium exists requires $\Omega(n^2)$ cc

Proof Alice x, Bob y $|x| = |y| = N = n^2$



Lemma computing whether a pure Nash equilibrium exists requires $\Omega(n^2)$ cc

Proof Alice x, Bob y $\quad |x| = |y| = N = n^2$



Extra rows/cols guarantee that only a cell (i^*, j^*) where both $X_{i^* j^*}$ and $Y_{i^* j^*} = 1$ is a pure Nash equilibrium.

(then a players best reply always has a value of 1 so a pure equilibrim requires a cell where both matrices have value 1.)

$\therefore$ Cost $o(n^2)$ solution to Nash $\Rightarrow$ cost $o(n^2)$ protocol for DISJ. #

2-Player ε-Nash is Hard

2 players. Each has an $N \times N$ payoff matrix

$$\begin{bmatrix} .3 & .6 & .5 \\ .2 & .4 & .1 \\ .9 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} .1 & .5 & .9 \\ .2 & .4 & .1 \\ 1 & .9 & 0 \end{bmatrix}$$

2-Player ε-Nash is Hard

2 players. Each has an $N \times N$ payoff matrix

$$A = \begin{bmatrix} .3 & .6 & .5 \\ .2 & .4 & .1 \\ .9 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} .1 & .5 & .9 \\ .2 & .4 & .1 \\ 1 & .9 & 0 \end{bmatrix}$$

$(\hat{x}, \hat{y})$ is an <u>ε-Nash Equilibrium</u> if:

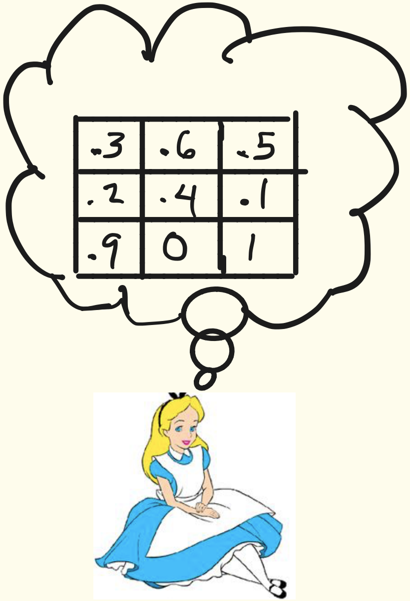
$$\hat{x}^T A \hat{y} \geq x^T A \hat{y} - \varepsilon \qquad \forall x$$

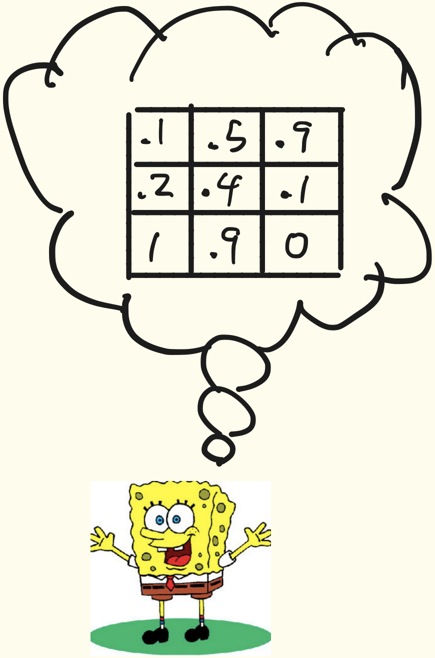
$$\hat{x}^T B \hat{y} \geq \hat{x}^T B y - \varepsilon \qquad \forall y$$

Finding ε-Nash Equilibrium is Hard

Theorem [Göös-Rubinstein '18]

The randomized communication complexity of finding an ε-Nash equilibrium is $\geq N^{2-o(1)}$



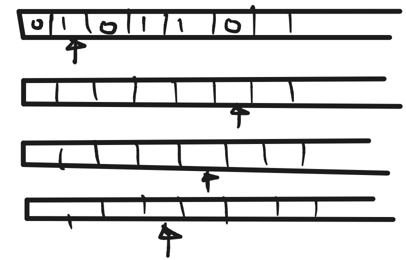


APPLICATIONS

- Streaming
- Property Testing
- game theory
- TIME/SPACE Turing Machine LBs
- Circuit complexity
- Proof complexity
- Extension Complexity
- Data Structures

TM TIME/SPACE LOWER BOUNDS

Multitape TMs : Read only input tape
plus $O(1)$ Read/Write tapes



Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$

We say that M recognizes/computes f if

$$\forall (x,y) \in \{0,1\}^{2n} \quad \begin{array}{l} f(x,y)=1 \Rightarrow M(x0^ny) = 1 \\ f(x,y)=0 \Rightarrow M(x0^ny) = 0 \end{array}$$

Theorem Let M compute f.

Then $$P^{cc}(f) \leq O\left(\frac{\text{Time}(M,n) \cdot \text{Space}(M,n)}{n}\right)$$

ie. if $P^{cc}(f) = \Omega(n)$ then any M computing f requires $\text{Time} \cdot \text{Space} = \Omega(n^2)$

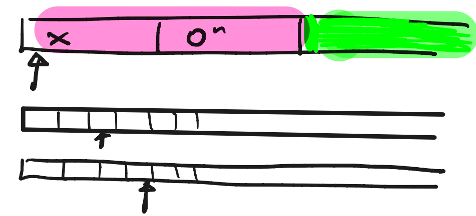
Proof

Let M be a TM that computes f in Time $T(n)$, space $S(n)$

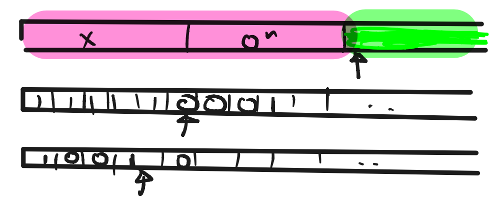
Then we will construct a CC protocol for f of cost $\leq T(n) \cdot S(n)$

Alice has x, Bob y.

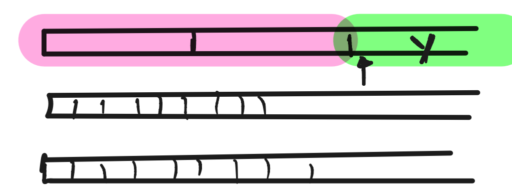
Alice simulates M on $x0^n$ until input head moves to green part



Then Alice sends entire content of R/W tape and head locations to Bob



Bob continues simulation with y on green part until input head moves to pink



$\vdots$

Comm. complexity:

\# of Rounds $= \frac{T(n)}{n}$ (since they have to spend n steps going thru middle zone)

cost per round $\leq O(S(n))$

$$\therefore \; CC(f) = O\left(\frac{T(n) \cdot S(n)}{n}\right)$$

<u>Note</u> 0^n in middle is kind of cheating

If we instead gave input $\boxed{x \mid y}$,

the cost of protocol would be $O(\text{\#-of-Reversals} * S(n))$