Last time: (basically) finished counting, approx. counting:
  • FPRAS for \#ONF (easy approx. counting problem);
  • no FPRAS for \#CYCLES unless P=NP (hard approx. counting problem)

Today: • 5-min sketch that any \#P can be efficiently approximated given an NP oracle
  • start unit on communication complexity AB13.13.2

Questions?

Final exam: take home

Last thing on counting:

\( (1 \pm \epsilon) \)-approx

Thm: Let \#P. There is an FPRAS \( \text{for } g \) that uses an NP oracle.

Sketch of 3 main ingr.:

1) Just worry abt \#3CNF (\#3CNF \#P-complete)
2) It's enough to get coarse \approx. alg.
   Getting \( \text{factor } 100 \approx. \Rightarrow FPRAS \#4 \)
3) Can use NP oracle + randomization to get a \( \text{factor } 100 \approx. \) alg. \#5

COMMUNICATION COMPLEXITY (cc)

Today: det c.c. of functions
Set up: 2 people A & B. Cooperating.
- Known f: X x Y \rightarrow \mathbb{Z} f(x, y) = x.
  (X = Y = \{0,1\}^n for us.)
- A has some x \in X, B has y \in Y
- They cooperatively communicate to compute f(x, y).

Q: how many bits do they need to communicate so that each of A, B can output right f(x, y) value? (ignore computation.)

All our ex: X = Y = \{0,1\}^n
x \times y \rightarrow \mathbb{Z} = \{0,1\}

Ex #1: any f: \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}
A can send all of x to B, B computes f(x, y), sends back to A. \(n+1\) is enough for any

Ex #2: X = Y = \{0,1\}^n
f(x, y) = PAR(x, y) = x_1 + \ldots + x_n + y_1 + \ldots + y_n \mod 2 \in \{0,1\}.
A can compute  = x_1 + \ldots + x_n \mod 2, send a to B
B output a + y_1 + \ldots + y_n \mod 2 \(2\) bits

Ex #3: X = Y = \{0,1\}^n; view x as subset of \[n\]
x = 1101001 \approx \text{set} \{1,2,4,7\}
y likewise.
Multiset x \cup y
f(x, y) = \text{median elt of} \[x \cup y\]. \ \mathbb{Z} = \{1, \ldots, n\}.
Binary search. Stage: interval where mid known to be is
\{i, i+1, \ldots, j\}. Let \( k = \frac{i+j}{2} \) be midpt.

\[ A \text{ sends } \left\{ \# \text{ els } x \text{ that are } \geq k \right\} \leq 2\log n \text{ bits} \]

\[ B \text{ uses this } \& \text{ explicit knowledge of } y \]

\[ \text{to det. whether } f(x, y) \geq k \text{ or } < k. \]

\[ B \text{ sends } "\#" \text{ or } "\&" \text{ bit Next stage.} \]

\[ \log n \text{ stages: total comm. } \leq (\log n) \cdot (2\log n + 1) = O(\log^2 n). \]

**Ex.** #4: \( X = Y = \{0, 1\} \) \( f(x, y) = EQ(x, y) = \left\{ \begin{array}{ll} 1 & x = y \\ 0 & x \neq y \end{array} \right. \)

B needs +1 bits: will show this.

---

**Formalization:** (det.) comm. protocols.

Protocol for \( f(x, y) \): complete set of rules for who says what \& when. A and B use only the \( f(x, y) \).

**Def:** A prot. for \( f: X \times Y \rightarrow \{0, 1\} \) is a bin. tree:

- each internal node: \( 2 \) children (0 \& 1);
- labeled either by \( (A, f_a: X \rightarrow \{0, 1\}) \) or \( (B, f_b: Y \rightarrow \{0, 1\}) \).

\( X = \{x_1, \ldots\} \)

**Diagram:**

```
A
\[ x_1 \rightarrow 0 \]
\[ x_2 \rightarrow 1 \]
\[ x_3 \rightarrow 1 \]
\[ x_4 \rightarrow 0 \]

B
\[ y_1 \rightarrow 1 \]
\[ y_2 \rightarrow 0 \]
```

A: \( x' \)

B: \( y' \)

\( A \) and \( B \) communicate through the tree.
Each leaf in tree labeled with some \( z \in \mathbb{Z} \).

Player walk prot. tree \( \tau \) execute it to enact prot., output = leaf they reach

\( \forall (x,y) \in X \times Y, \text{ leaf reached on } (x,y) \text{ is } z = f(x,y). \)

**Ex:** translate ex 1-4 into formalism.

**Def:** Cost of prot. \( P = \text{depth of prot. tree} = \text{worst-case # bits A, B could communicate across all } (x,y). \)

**Def** Def. comm. complexity of \( f : X \times Y \rightarrow \mathbb{Z} \) written \( D(f) \), is min cost of any prot. \( P \) for \( f \).

**View** \( f : X \times Y \rightarrow \mathbb{Z} \) as a matrix

\[ X \rightarrow Y \]

"comm. mtx of \( f \)"

**Key insight:**
**Def**: A "rectangle" in $X \times Y$ is a subset $R \subseteq X \times Y$ s.t. $R = U \times V$ some $U \subseteq X, V \subseteq Y$.

Note $R \subseteq X \times Y$ is a rect. iff

- $\forall x_1, x_2, y_1, y_2 : (x_1, y_1) \in R \Rightarrow (x_1, y_2) \in R$
- $(x_2, y_1) \in R \Rightarrow (x_2, y_2) \in R$

**Notation**: Given $\text{prot } P$ and node $v$ in tree $P$, write $R_v$ to denote $\{ (x, y) : (x, y) \text{ reach node } v \}$.

Every $(x, y)$ reaches some leaf of $P$.

- So $\{ R_v : \text{l a leaf} \}$ partitions $X \times Y$ into...
disjoint subsets.

\[ R_{l_1} \quad R_{l_2} \quad \_ \quad \_ \quad \_ \quad \_ \quad l_b \]

- For any node \( n \) in prot tree \( P \), \( R_n \) is a rectangle.

  Induction: base case: root

  \[ R_{\text{root}} = X \times Y \]

\[ X'_0 = \{ x \in X' : f_n \text{ at node } w \text{ outputs } 0 \} \]

\[ X'_0, X'_1, \text{ disjoint}; \quad X'_0 \cup X'_1 = X' \]

So... \( R_{l_b} \) is a rect. for each leaf \( l \) of \( P \).

For each fixed \( l \), \( f(xy) \) is same across all \( (x,y) \) that reach \( l \) (b/c \( P \) is a prot. for \( f \)).

So each \( R_{l_b} \) (l leaf) is "f-monochromatic" rectangle.
Summarize:

- Given a prot. $P$ computing $f(x,y)$, leaves of $P$ induce partition of $X \times Y$ into $f$-monochr. rect's.

- # rect's in partition = # leaves in prot.

- So... if every part. of $X \times Y$ into $f$-monochr. rect. requires $\geq t$ rect,
  then every prot. for $f$ must have $\geq t$ leaves...
  so depth of prot. must be $\geq \log_2 t$
  hence $O(f) \geq \log_2 t$.

$\text{Ex: } X = Y = \{0,1\}^n, f(x,y) = EQ(x,y) = \left\{ \begin{array}{ll} 1 & x=y \\ 0 & x \neq y \end{array} \right.$

$X = \{0,1\}^n$

Every 1 must be in its own $1 \times 1$ rect.
So need $2^n$ rect.
just for the 1's;

\[ \text{so \ total \ # \ rectangles \ in \ any \ } f-\text{monochrom. \ part. \ is} \]
\[ > 2^n + 1 \quad \text{so \ # \ leaves \ in \ prefix \ tree} > 2^n + 1 \]
\[ \text{so \ depth} > \log (2^n + 1) > n \]
\[ \text{so \ depth} \geq n + 1. \quad D(\text{EQ}_n) = n + 1 \]

Next time: Time/Space tradeoffs for TMs:

any TM for palindromes must have

\[ T(n) \cdot S(n) \geq \Omega (n^2) \]

\[ \text{time} \uparrow \quad \text{space} \uparrow \]

Rand. c.c.,

lbs from that,  
cut exit...