DETERMINING IF \( X = Y \)
1. Alice has $x$, Bob has $y$.
2. They want to see if $x = y$ communicating as few bits as possible.
3. We call this problem EQ.
OBVIOUS PROTOCOL

1. Alice has $a_1 \cdots a_n$. Bob has $b_1 \cdots b_n$. 
2. Alice sends $a_1 \cdots a_n$ to Bob (n bits). 
3. Bob compares $a_1 \cdots a_n$ to $b_1 \cdots b_n$. 
   If equal send 1, else send 0. (1 bit.) 

So \( \text{EQ} \) can be solved with \( n + 1 \) bits.
1. EQ **REQUIRES** $\sim n$ bits.
2. Can do EQ **with** $\sim \sqrt{n}$ bits, but no better.
3. Can do EQ **with** $\sim \log n$ bits, but no better.
EQ \textbf{REQUIRES} \( n + 1 \) bits.
So, for Alice and Bob to determine if two \( n \)-bit strings are equal \textbf{REQUIRES} \( n + 1 \) bits.
(Proven by Andrew Yao in 1979.)
What if we

1. Allow Alice and Bob to flip coins, and
2. allow a probability of error $\leq \frac{1}{n}$. 
1. Alice has $a_1 \cdots a_n$. Bob has $b_1 \cdots b_n$.
2. Alice rand $S \subseteq \{1, \ldots, n\}$, $|S| = 10$.
3. For $i \in S$ Alice sends $(i, a_i)$. 10 log $n$ bits.
4. For each $(i, a_i)$ that Bob checks “$a_i = b_i$?”.
5. If always YES, Bob sends 1, else sends 0.
1. Protocol is $\sim \log n$ bits. **GOOD!**
2. Prob of error $\to 1$ as $n \to \infty$. **BAD!**
3. Does well if input is unif chosen. **GOOD!**
4. Not really what we want. **BAD!**
5. KEY PROBLEM: Protocol too local.
LESS NAIVE IDEA

1. Alice has $a_1 \cdots a_n$. Bob has $b_1 \cdots b_n$.

2. Alice computes $a_1 + \cdots + a_n$.
   Sends 1 if sum is ODD
   Sends 0 if sum is EVEN.

3. Bob computes $b_1 + \cdots + a_n$.
   If PARITY agrees then send 1 (EQUAL)
   else 0 (NOT EQUAL)
1. Only send $\sim 1$ bit. **GOOD.**
2. Bit sent uses ALL of $a_1 \cdots a_n$. **GOOD.**
3. Protocol will be wrong alot. **BAD.**
4. Speculation: Can we use $a_n + \cdots + a_1$ remainder when divided by $3$? $4$? $5$?
$a = b \pmod{c}$ means

1. $a/c$ and $b/c$ have same remainder.
2. $b \in \{0, 1, \ldots, c - 1\}$.

EXAMPLES:

1. Any odd number is $\equiv 1 \pmod{2}$.
2. $100 \equiv 9 \pmod{7}$ since $100 = 7 \times 13 + 9$
1. Everyday Example: Clock Arithmetic
2. For all $m$ you can do $\div$, $-$, $\times$ in $\mathbb{Z}_m$.
3. For $m$ prime you can also do $\div$ in $\mathbb{Z}_m$. 

$\mathbb{Z}_m = \{0, 1, \ldots, m-1\}$ and all of the operations are mod $m$. 

Bill Gasarch- U.of MD-College Park gasarch@cs.umd.edu
1. If $f$ is a polynomial over the reals of degree $d$ then $f$ has at most $d$ roots.

2. If $f$ is a polynomial over the complex numbers of degree $d$ then $f$ has at most $d$ roots. ($d$ if you count multiplicities.)

3. Let $p$ be a prime. If $f$ is a polynomial over $\mathbb{Z}_p$ of degree $d$ then $f$ has at most $d$ roots.
1. Alice has $a_0 a_1 \cdots a_{n-1}$. Bob has $b_0 b_1 \cdots b_{n-1}$. Alice sends Bob a prime $p$, $n^2 \leq p \leq 2n^2$. 

2. Alice picks $z \in \{1, \ldots, p - 1\}$ RAND. 
   Alice computes, mod $p$, 
   $$y = a_0 + a_1 z + a_2 z^2 + \cdots + a_{n-1} z^{n-1}$$ 
   Alice sends $(z, y)$ to Bob. 

3. Bob computes, mod $p$, 
   $$y' = b_0 + b_1 z + b_2 z^2 + \cdots + b_{n-1} z^{n-1}$$ 
   If $y = y'$ then send 1, else send 0.
1. Protocol exchanges $\sim \log n$ bits.
2. Prob of error is $\leq \frac{1}{n}$.
   **WHY:** If there is an error then $z$
   is a root of the poly $a(x) - b(x)$
   There are only $n$ such roots so the probability
   of this is very low.
3. This result is due to Melhorn and Schmidt, 1982.
COMMUNICATION COMPLEXITY
by
Kushilevitz and Nisan.