GEOGRAPHY is PSPACE-complete

- We are reducing from the language:
  \[ \text{QSAT} = \{ \phi : \phi \text{ is a 3-CNF, and} \exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \cdots \phi(x_1, x_2, x_3, \ldots, x_n) \} \]

  to the language:
  \[ \text{GEOGRAPHY} = \{(G, s) : G \text{ is a directed graph and player I can win from node } s\} \]

**Extended Church-Turing Thesis**

- The belief that TMs formalize our intuitive notion of an efficient algorithm is:

  - The "extended" Church-Turing Thesis:
    everything we can compute in time \( t(n) \)
    on a physical computer can be computed on a Turing Machine in time \( t(n)^{O(1)} \) (polynomial slowdown)

  - Randomized computation challenges this belief
Randomness in computation

- Example of the power of randomness
- Randomized complexity classes

Communication complexity

two parties: Alice and Bob
function $f(0,1)^n \times (0,1)^n \rightarrow (0,1)$
Alice holds $x \in (0,1)^n$; Bob holds $y \in (0,1)^n$

- Goal: compute $f(x, y)$ while communicating as few bits as possible between Alice and Bob
- count number of bits exchanged (computation free)
- at each step: one party sends bits that are a function of held input and received bits so far

Communication complexity

- simple function (equality):
  $EQ(x, y) = 1$ iff $x = y$
- simple protocol:
  - Alice sends $x$ to Bob ($n$ bits)
  - Bob sends $EQ(x, y)$ to Alice (1 bit)
  - total: $n + 1$ bits
  - (works for any predicate $f$)

Communication complexity

- Can we do better?
  - deterministic protocol?
  - probabilistic protocol?
    - at each step: one party sends bits that are a function of held input and received bits so far and the result of some coin tosses
    - required to output $f(x, y)$ with high probability over all coin tosses

Theorem: no deterministic protocol can compute $EQ(x, y)$ while exchanging fewer than $n+1$ bits.

- Proof:
  - "input matrix":
    $X = (0,1)^n$
    $Y = (0,1)^n$
    $f(x,y)$
    inputs $x$ causing $A$ to send 1
    inputs $x$ causing $A$ to send 0
March 6, 2023
CS21 Lecture 25

**Communication complexity**

- B sends 1 bit depending only on y and received bit:

  - X = \{0,1\}^n
  - Y = \{0,1\}^n

- Inputs y causing B to send 1
- Inputs y causing B to send 0

---

**Communication complexity**

- At end of protocol involving k bits of communication, matrix is partitioned into at most \(2^k\) combinatorial rectangles
- Bits sent in protocol are the same for every input \((x, y)\) in given rectangle
- Conclude: \(f(x, y)\) must be constant on each rectangle

---

**Communication complexity**

- Any partition into combinatorial rectangles with constant \(f(x, y)\) must have at least \(2^n + 1\) rectangles
- Protocol that exchanges \(n\) bits can only create \(2^n\) rectangles, so must exchange at least \(n+1\) bits.

---

**Communication complexity**

- Protocol for EQ employing randomness?
  - Alice picks random prime \(p\) in \(1...4n^2\), sends:
    - \(p\)
    - \((x \mod p)\)
  - Bob sends:
    - \((y \mod p)\)
  - Players output 1 if and only if:
    - \((x \mod p) = (y \mod p)\)

---

**Communication complexity**

- \(O(\log n)\) bits exchanged
- If \(x = y\), always correct
- If \(x \neq y\), incorrect if and only if:
  - \(p\) divides \(|x - y|\)
  - \# primes in range is \(\geq 2n\)
  - \# primes dividing \(|x - y|\) is \(\leq n\)
  - Probability incorrect \(\leq 1/2\)

Randomness gives an exponential advantage!!
Communication complexity

Goal: compute \( f(x, y) \) while communicating as few bits as possible between Alice and Bob

Example: \( \text{EQ}(x, y) = 1 \) if \( x = y \)

• Deterministic protocol: no fewer than \( n+1 \) bits
• Randomized protocol: \( O(\log n) \) bits

Extended Church-Turing Thesis

• Common to insert “probabilistic”:

The “extended” Church-Turing Thesis
everything we can compute in time \( t(n) \) on a physical computer can be computed on a probabilistic Turing Machine in time \( t(n)^{O(1)} \) (polynomial slowdown)

Randomized complexity classes

• model: probabilistic Turing Machine
  – deterministic TM with additional read-only tape containing “coin flips”

  input tape

  finite control
  read/write head

  read head

  ...