Definitions

- We say a (deterministic) protocol $P$ computes $f$ if $\forall (x, y) \in \{0, 1\}^n \times \{0, 1\}^n, P(x, y) = f(x, y)$
- A randomized protocol $P$ computes $f$ with error probability $\varepsilon$ if $\forall (x, y) \in \{0, 1\}^n \times \{0, 1\}^n, \Pr[P(x, y) \neq f(x, y)] \leq \varepsilon$
- We define $\text{cost}(P) = \max_{x, y} \text{number of bits communicated by } P(x, y)$
- Also, we define $D(f)$ to be the minimum cost over all deterministic protocols computing $f$

Randomized Max-Two

Consider the following (deterministic) algorithm to find the largest two elements in an input:

```plaintext
function Max-Two(S)
    n ← |S|
    $(m_1, m_2) = (\max(S[0], S[1]), \min(S[0], S[1]))$
    for $i = 2, 3, \ldots, n - 1$ do
        if $S[i] < m_2$ then
            continue
        end if
        if $S[i] < m_1$ then
            $(m_1, m_2) = (m_1, S[i])$
            continue
        end if
        $(m_1, m_2) = (S[i], m_1)$
    end for
    return $(m_1, m_2)$
end function
```

Suppose we run this algorithm on a uniformly random permutation of the array $[1, 2, \ldots, n]$. What is the expected number of total comparisons we’ll make?

The $X_i$ be the indicator RV st, $X_i = 1$ if during the $i$th iteration, we reach the second comparison, and $X_i = 0$ otherwise. We first compute $E[X_i]$.

$$E[X_i] = Pr[X_i = 1] = \frac{2}{i + 1}$$

Since we will need two comparisons if $X_i$ is either the largest or second largest element so far. Let $Y$ be the RV denote the total number of comparisons, we have

$$E[Y] = E[1 + \sum_{i=2}^{n-1} (X_i + 1)] = n - 1 + 2 \sum_{i=2}^{n-1} \frac{1}{i + 1} = n - 4 + 2H_n$$
**Communication in Rectangles**

Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a function, and define the matrix $M_f \in \{0,1\}^{2n \times 2n}$ such that $(M_f)_{x,y} = f(x,y)$. We define $R \in X \times Y$ to be a rectangle if $R = A \times B$ for some $A \subseteq X$ and $B \subseteq Y$. Further, we say $R$ is $f$-monochromatic if $f$ takes on a constant value over inputs in $R$.

(a) Let $R$ be any rectangle. Show that if $(a,b), (c,d) \in R$, then $(a,d), (b,c) \in R$.

(b) Suppose that a partition of $X \times Y$ into $f$-monochromatic rectangles requires $2^t$ rectangles. Show that $D(f) \geq t$.

(c) Prove that $D(\text{EQ}) = \Omega(n)$.

---

(a) We know that $R = A \times B$. Since $(a,b), (c,d) \in R$, this means that $a, c \in A$ and $b, d \in B$. This means that $(a,d), (c,b) \in R$.

(b) We think about the information that we can get from reading a transcript of length $c$ of the communication between Alice and Bob.

Before the communication begins, having not seen Alice or Bob’s information, as far as we know, $(x, y)$ could lie anywhere in the matrix $M_f$. Consider this to be our working rectangle.

Now, suppose Alice sends Bob one bit of information. Since Alice can only send information based on $x$, we know that this information can only tell us about the row that $(x, y)$ is in. This splits the possible rows into at most two parts based on what bit she sends. After seeing this bit, we can narrow $(x, y)$ down to one of two possible rectangles.

Now suppose that Bob sends a bit. This can only tell us about the column that $(x, y)$ is in. We want to narrow down our working rectangle. Given the previous working rectangle, and Bob’s bit, there are only two possible working rectangles that we could go to.

Now, note that it does not matter who sends the bits in what order. Each successive bit that is sent can split the working rectangle into at most two possibilities. This means that after reading $c$ bits, there are at most $2^c$ possible rectangles that we could have narrowed $(x, y)$ down to.

For this protocol to work correctly, we argue that the protocol must result in narrowing $(x, y)$ down to a monochromatic rectangle. Suppose that reading the transcript does not result in a monochromatic rectangle. This means that there are two points in the working rectangle with different values of $f$.

They cannot be in the same row, as then Alice will have no way of telling which of the points is actually $(x, y)$ by the transcript that was sent. Symmetrically, they cannot be in the same column, or Bob would not know the answer.

Suppose they are in different rows and different columns. Then, there must be an index of the matrix that shares a row with one, and a column with the other. By part (a), this is part of the same rectangle. This index will result in the same issue shown in the last paragraph.

Finally, since the protocol can differentiate between at most $2^c$ rectangles, and there are $2^t$ possible monochromatic rectangles that we need to tell apart, we have that $2^{D(f)} \geq 2^t$, so:

$$D(f) \geq t$$

(c) If we consider EQ in this setting, we know that it has at least $2^n$ rectangles: one for each possible $(a,a) \in X \times Y$. This means that the communication complexity is at least $n$ by part (b).
**Linear Algebra Meets Rectangles**

Prove that

\[ D(f) \geq \log(\text{rank}(M_f)). \]

Recall (or, learn) that the rank of a matrix is defined to be the maximum number of linearly independent column vectors in the matrix.

(Hint: Recall that rank is sub-additive, i.e. that \( \text{rank}(A + B) \leq \text{rank}(A) + \text{rank}(B) \).)

For a rectangle \( R \), define its corresponding matrix \( M_R \) such that

\[
(M_R)_{x,y} = \begin{cases} 
1 & (x,y) \in R \\
0 & \text{otherwise},
\end{cases}
\]

and observe that \( \text{rank}(M_R) \leq 1 \) for any \( R \). Now, suppose that \( D(f) = c \), and let \( P \) be a minimum-cost protocol. Define \( R(P) \) to be the set of rectangles that partition \( f^{-1}(1) \), and observe that \( |R(P)| \leq 2^c \) (this follows from (b) of the previous problem). Now, we have

\[
M_f = \sum_{M_R \in R(P)} M_R,
\]

and hence

\[
\text{rank}(M_f) \leq \sum_{M_R \in R(P)} \text{rank}(M_R) \leq 2^c.
\]

Taking the log of both sides yields the desired result.

**(EXTRA) PITiful polynomials**

Consider the PIT problem: given as input a polynomial, written using any of

\[ \Sigma = \{(,)+,\} \cup \{x : i \in \mathbb{N}\} \cup \mathbb{Q}, \]

calculate whether the polynomial is equal to 0.

Example input:

- \((x_1 x_1 x_1 + x_3)(x_5 + x_1)\) (which is not 0).
- \((x_1 + x_2)(x_1 - x_2) - x_1 x_1 - x_2 x_2\) (which is 0).

(a) Before we solve this problem, we need a lemma which you might find helpful.

**Lemma 1 (Schwartz-Zippel)** If \( P \) is a non-zero polynomial on variables \( x_1, \ldots, x_n \), and is of degree at most \( d \), then if we draw each \( x_i \) uniformly from any set \( S \subseteq \mathbb{R} \),

\[
\Pr[P(x_1, x_2, \ldots, x_n) = 0] \leq \frac{d}{|S|}.
\]

Remark: You can think of this lemma as a kind of multivariable fundamental theorem of algebra.

Hint: You can prove this by induction on \( n \). You probably know the base case, and don’t forget FToA in the inductive step.
We prove this by induction on \( n \).

**Base Case** \( n = 1 \). This follows from the fundamental theorem of algebra, which says the polynomial can have at most \( d \) roots overall.

**Inductive Step.** Assume that every polynomial \( P' \) on \( n-1 \) variables with degree \( d' \) has \( \Pr[P'(x_1, x_2, \ldots, x_{n-1}) = 0] \leq d'/|S| \). 

Rewrite our polynomial \( P \) grouping the \( x_n \) terms, so in the end we get an expression of the form

\[
P(x_1, \ldots, x_n) = \sum_{i=1}^{d} x_n^i Q_i(x_1, \ldots, x_{n-1}).
\]

Since \( P \) is non-zero, some \( Q_i \) is non-zero. Fix \( i \) to be the largest with this property. Now fix values \( x^*_1, x^*_2, \ldots, x^*_{n-1} \), all except \( x_n \), with \( Q_i(x^*_1, \ldots, x^*_{n-1}) \neq 0 \). By union bound we have

\[
\Pr[P(x^*_1, \ldots, x^*_{n-1}, x_n) = 0] \leq \Pr[Q_i(x_1, \ldots, x_{n-1}) = 0] + \Pr[P(x^*_1, \ldots, x^*_{n-1}, x_n) = 0],
\]

if we take the randomness in the second probability over \( x_n \) only, i.e., treating the \( x^*_1, \ldots, x^*_{n-1} \) as fixed. We can also think of this step as conditioning over all possible values of \( x_1, \ldots, x_{n-1} \).

By the induction hypothesis, the first quantity is upper-bounded by \( (d-i)/|S| \), since \( Q_i \) can have degree at most \( d-i \). The second probability is upper-bounded by the probability if we take \( x^*_1, \ldots, x^*_{n-1} \) as fixed, and take the randomness just over \( x_n \). In this way, we just need to calculate \( \Pr[P(x^*_1, \ldots, x^*_{n-1}, x_n) = 0] \) for fixed \( x^*_1, \ldots, x^*_n \). By the fundamental theorem of algebra, this polynomial in \( x_n \) has at most \( i \) roots, since it is of degree at most \( i \). Therefore

\[
\Pr[P(x_1, \ldots, x_{n-1}, x_n) = 0] \leq i/|S|.
\]

Combining these two bounds into (1) yields an upper bound of \( d/|S| \), completing the proof.

(b) Come up with an efficient randomized algorithm to solve this problem with error probability \( \varepsilon \).

(Hint: Schwartz-Zippel Lemma)

```
function PIT(P)
    Calculate the degree d of the polynomial.
    Pick S = {1, ..., 2d}.
    Choose N such that 1/2^N < \varepsilon (N > \log_2(1/\varepsilon)).
    for i in [1..N] do
        Pick x_1, ..., x_n each i.i.d. uniformly from S.
        if P(x_1, ..., x_n) \neq 0 then
            return False
        end if
    end for
    return True
end function
```

By Schwartz-Zippel, the probability the algorithm makes a mistake on any iteration is at most \( 1/2 \), so the probability we make a mistake on any iteration is at most \( 1/2^N < \varepsilon \).

The algorithm is poly-time because \( N \) is polynomial in the input, and evaluating \( P \) is polynomial time given \( x_1, \ldots, x_n \).
(c) Something to ponder: some people believe that randomization gives you no more power than determinism. If we believe them, then try to come up with a deterministic algorithm to solve this problem.