
1: Isomorphic Graphs

Problem Statement:

Alice and Bob each have an undirected graph on n vertices. How much communication do they need to find out deterministically if their graphs are isomorphic?

Analysis:

The problem of Graph Isomorphism by virtue of being an equivalence relation very similar to that of equality. In particular, it is easy to construct a fooling set of size equal to the number of equivalence classes. Hence it remains to show that there exist

Definitions:

Let \mathbf{G}_n = the set of unique (up to isomorphisms) simple undirected graphs on n vertices

Let $IS_n : \mathbf{G}_n \times \mathbf{G}_n \rightarrow \{0, 1\}$ be the graph isomorphism function i.e. 1 if the Graph x is an isomorphism of Graph y and 0 otherwise

Claim.

$$D(IS_n) \geq \frac{n^2}{2} - \Omega(n(\log n))$$

Proof. Number of unique simple undirected graphs (allowing for double counting of graphs that are isomorphic to each other) on n vertices = $2^{\binom{n}{2}}$

But every graph is isomorphic to at most $n!$ other graphs

$$\text{Thus } |\mathbf{G}_n| \geq \frac{1}{n!} \cdot 2^{\binom{n}{2}}$$

$$\text{Since the function is an equivalence relation } f_s(IS_n) = |\mathbf{G}_n| \geq \frac{1}{n!} \cdot 2^{\binom{n}{2}}$$

$$\text{Thus } D(IS_n) \geq \log\left(\frac{1}{n!} \cdot 2^{\binom{n}{2}}\right) = \binom{n}{2} - \log(n!) = \frac{n^2}{2} - \Theta(n \log(n))$$

This lower bound is tight as there is a protocol with this cost. In particular, if Alice and Bob (being computationally unbounded entities) simply match the graph they receive to a graph $\in \mathbf{G}_n$ then running equality on the $(\frac{n^2}{2} - \Theta(n \log(n)))$ bit representation of the graphs will be sufficient for computing IS_n

2: Integer Multiplication

Problem Statement:

How much deterministic communication does it take to compute the n least significant bit of the product of two natural numbers, of which Alice has one and Bob the other?

Analysis:

The first simplification that can be done on this problem is reducing the problem from multiplication of any two natural numbers to the multiplication of the integers represented by 2 n -bit strings (that are in particular the least significant n -bits of the original natural number's representation) as any bit more significant cannot affect the output of this function.

Definitions:

Let $Product_n : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be the function on n bits strings indicating the n th least significant bit of the product of the integers represented by the strings

Claim.

$$D(Product_n) \geq \log(n)$$

Proof. This proof uses the rank bound technique and shows that the characteristic matrix of the function $M_{Product_n}$ has rank $\geq n$ to arrive at the lower bound for the communication complexity of the function.

Assuming rows and columns are arranged in lexicographic order it easy to see that the rows corresponding to powers of 2 $\in [0, 2^n]$ are linearly independent.

An inductive argument proves this by showing that for any $k < 2^n$ the row corresponding to k th power of 2 must be linearly independent of the previous $k - 1$ rows corresponding to powers of 2 as the first $2^n / 2^{k-1}$ numbers have 0s in their $k - 1$ most significant bits thus must have 0s the columns corresponding to them in all the previous $k - 1$ rows but necessarily half of them must have a 1 in the k th most significant bit and thus will have 1s in the row corresponding to 2^k , thus this row is linearly independent of the previous $k - 1$ rows (corresponding to powers of 2). A similar argument works for the base case in showing that rows corresponding to 2^0 and 2^1 are linearly independent.

$$\text{Thus } \text{rank}(M_{Product_n}) \geq \log(2^n) = n$$

$$\text{Therefore by the rank bound } D(f) \geq \log(\text{rank}(M_{Product_n})) \geq \log(n)$$

Claim.

$$D(Product_n) = n + 1$$

Proof. Proving first $\dim(\text{kernel}(M_{Product_n})) < 2^{n-1}$

Conjecture: Vectors of the following form completely span the null space: 1 in the a^{th} and b^{th} positions and -1 in the c^{th} and d^{th} positions where a and b are such that $a + b = 2^n$, $c = 2^{n-1} - 1$ and $d = 2^{n-1} + 1$ positions

I was unable to prove this conjecture (perhaps because this is a consequence of a deeper phenomenon), however the intuition behind this relies on the following argument. Moreover, there

was overwhelming evidence from empirical data that conformed to this hypothesis (the appendix contains the python script used in order to obtain this empirical data):

The vectors in the nullspace are a subset of the linear combinations of the vectors that represent in essence multiplication by 0. This is to say that the vectors of the form mentioned above are vectors where the 4 indices with non-zero values along with values sum up to 0 i.e.

$a + b - c - d = 0$. The choice for 4 values of this form is not arbitrary as it can be argued easily that any other vector of such a form in the nullspace can be expressed as a linear combination of these vectors, and the choice of 4 in particular allows for easy analysis of the possible vectors of this type. Since $a \in [1, 2^{n-1} - 2]$ the number of vectors of this form are strictly less than 2^{n-1} and thus $\dim(\text{kernel}(M_{\text{Product}_n})) < 2^{n-1}$

By the rank nullity theorem, this implies $\text{rank}(M_{\text{Product}_n}) > 2^{n-1}$

Thus by the rank bound, $D(\text{Product}_n) \geq \lceil \log(\text{rank}(M_{\text{Product}_n})) \rceil + 1 \geq n + 1$

3: Boolean Formulas

Problem Statement:

A Boolean formula in variables z_1, \dots, z_n is a fully parenthesized expression with operands $z_1, \neg z_1, \dots, z_n, \neg z_n$ and operators \wedge and \vee . Let $\phi(z_1, \dots, z_n)$ be a Boolean formula in which every variable occurs exactly once. Prove that computing $\phi(x \oplus y)$ deterministically on input $x, y \in \{0, 1\}^n$ requires $\Omega(n)$ bits of communication.

Analysis:

The requirement that every operand appear exactly once ensures dependence on every operand by eliminating all redundancy from the function and as such this observation can be used to justify a tight lower bound.

Definitions:

Let $Boolean_n : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ be any boolean formula of the form described in the problem statement.

Let $z = x \oplus y$ and z_i be the i^{th} bit of z

Claim.

$$D(Boolean_n) = n + 1$$

Proof. \exists Two possible values of z s.t. that they differ only in z_i but do not have the same output $\forall i$

This follows from every operand appearing exactly once as in the 2^n possible values for z , there must exist two strings with $n - 1$ bits with values such that the value of the formula depends entirely on z_i (as the $n - 1$ bits can take all possible values and since the formula cannot contain any constants, the formula with values substituted in for the $n - 1$ bits cannot be constant and in particular can be reduced to either z_i or $\neg z_i$)

Hence any protocol must partition the elements in the characteristic corresponding to these values.

But since this is true $\forall i$ there exist n such distinct¹ partitioning cuts.

Any valid protocol must induce these n partitions (as for any given bit, there exists x and y such that it is impossible for Alice or Bob to know precisely which side of the corresponding partition the final answer lies in solely on the basis of x_i or y_i respectively²)

To induce these n distinct partitions any protocol must send at least n bits.

Moreover, since the boolean formula cannot contain constants and due to the aforementioned observations, every boolean formula of this form must also necessarily be non-constant.

$$\text{Hence } D(Boolean_n) = n + 1$$

¹Note that the partitions must necessarily be distinct as they are not this implies that the 2 strings corresponding to bit i differ in bit j as well - contradiction

²The characteristic matrix for the xor function with full rank evidences this

4: Jazzy Inner Product

Problem Statement:

Define $f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ by $f(x, y) = 1 \iff \sum x_i \cdot y_i = 0 \pmod{18181}$. What is the nondeterministic communication complexity of f ?

Analysis:

This problem seems identical to the analysis for modulo 2 Inner Product function discussed in class and since the 18181 is a prime number, it allows for a similar analysis of rectangle size.

Definitions:

Let $JIP_n : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ be the function f described in the problem statement

Claim.

$$N(JIP_n) = \Omega(n)$$

Proof. Note that in any 1-monochromatic rectangle $R = A \times B$, A and B can be interpreted as subspaces³ in \mathbb{F}_{18181} and therefore can be enlarged to $R' = \text{span}(A) \times \text{span}(B)$

Let $A' = \text{span}(A)$ and $B' = \text{span}(B)$

Let $\dim(A') = k$ and since B' is orthogonal to A' in \mathbb{F}_{18181} , it can have dimension at most that of the orthogonal complement of A' and thus $\dim(B') \leq n - k$

Therefore $|A'| = \text{number of boolean vectors in } A' = 2^k$

Similarly, $|B'| = \text{number of boolean vectors in } B' \leq 2^{n-k}$

Therefore $|R'| = |A'| \cdot |B'| \leq 2^n$

Thus choosing $\mu = \text{the uniform distribution over } f^{-1}(1)$

$$\mu(R') \leq \frac{2^n}{2^n + 2^{n-1} \cdot (2^n - 1)} < \frac{1}{2^{n-1}} \forall R' \text{ (1-monochromatic rectangles)}$$

$$\text{Thus } RS(JIP_n) = \frac{1}{2^{n-1}}$$

Hence by the rectangle size bound $N(JIP_n) = \lceil \log(2^{n-1}) \rceil \geq n$

Thus $N(JIP_n) = \Omega(n)$

³in the finite field with 18181 elements

5: Relative Primality

Problem Statement:

Alice and Bob's inputs are integers a and b , respectively, where $a, b \in [1, 2^n]$. Prove that $\Theta(n/\log n)$ bits are necessary and sufficient to verify nondeterministically that a and b are relatively prime.

Analysis:

...

Definitions:

Let $RP_n : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$

Claim.

$N(RP_n) = \Omega(n/\log(n))$

Proof. Use prime number theorem

Not really sure how to progress from there, going to write program to see matrices, but need to come up with protocol in order to prove upper bound

6: Orthogonal Subspaces

Problem Statement:

On input linear subspaces $A, B \subseteq \mathbb{F}_2^n$, prove that $\Theta(n^2)$ bits of nondeterministic communication are necessary and sufficient to check if A and B are orthogonal.

Analysis:

Using the understanding that the orthogonal complement of a subspace is the subspace that contains all orthogonal vectors there is an intuitive way to construct a large fooling set.

Definitions:

Let $OS_n : \{0, 1\}^{n^2} \times \{0, 1\}^{n^2} \rightarrow \{0, 1\}$

Let A be any subspace in the set of valid inputs

Claim.

$$N(OS_n) = \Omega(n^2)$$

Proof. A valid fooling set is the set of tuples of the form $(A, A^\perp) \forall A$

By definition of orthogonal complement, it is the subspace containing **all** vectors orthogonal to a subspace, hence one of the crosspoints for any two points must necessarily be 0 as atleast one of the crosspoints cannot correspond to a subspace of the orthogonal complement.

Thus the $f_s(OS_n) = |\text{Subspaces of } \mathbb{F}_2^n|$

Number of subspaces of $\mathbb{F}_2^n = 1 + \binom{2^n}{n-1} + \dots + \binom{2^n}{1} + 1$

The largest term in the above expression is by far the second term as the terms representing combinations reduce in magnitude as $n \ll 2^{n-1}$

$$\therefore f_s(OS_n) = \Omega\left(\binom{2^n}{n-1}\right) = \Omega(2^{n^2})$$

$$\text{Thus } N(OS_n) = \Omega(\log(2^{n^2})) = \Omega(n^2)$$

The trivial protocol shows that this bound is tight.

7: Communication v/s Randomness

Problem Statement:

Prove that any randomized protocol for EQn with probability of correctness $2/3$ and communication cost c must use more than $\log_2(n/c)$ bits of randomness.

Analysis:

say something

Definitions:

Let $EQ_n : \{0, 1\}^{n^2} \times \{0, 1\}^{n^2} \rightarrow \{0, 1\}$

Let P be a particular randomized protocol for EQ_n with cost c with probability of correctness = $2/3$

Let B_P = number of random bits used by protocol P

Claim.

$B_P \geq \log(n/c)$

Proof.

8: Better Than Random

Problem Statement:

Prove that every $f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ has a randomized protocol with constant cost and error at most $1/2 - \Theta(2^{-n/2})$

Analysis:

Thoughts

Definitions:

Let $EQ_n : \{0, 1\}^{n^2} \times \{0, 1\}^{n^2} \rightarrow \{0, 1\}$

Let P be a particular randomized protocol for EQ_n with cost c with probability of correctness = $2/3$

Let B_P = number of random bits used by protocol P

Claim.

$B_P \geq \log(n/c)$

Proof.

Appendix: Python Script used for Empirical Data for Rank in Various Fields

```

from sage.all import *
num_of_bits = int(raw_input("How many bits?\n"))
m_f = []
for x in range(0, 2**num_of_bits):
    l1 = []
    for y in range(0, 2**num_of_bits):
        z = x * y
        binary = ("{0:b}".format(z))[: -1]
        if (len(binary) < num_of_bits):
            l1.append(0)
        else:
            l1.append(int(binary[num_of_bits - 1]))
    m_f.append(l1)
field_num = 2** (num_of_bits - 1)
print(2*field_num)
matrix_f = (matrix(GF(2), m_f))
print(matrix_f.rank())
matrix_f = (matrix(GF(2**num_of_bits), m_f))
print(matrix_f.rank())
matrix_f = (matrix(GF(2** (num_of_bits+1)), m_f))
print(matrix_f.rank())
matrix_f = (matrix(m_f))
print(matrix_f.rank())
kernel_f = (kernel(matrix_f)).matrix()
print(kernel_f)
hypothesis = True
row_num = 0
for row in kernel_f:
    pos = 0
    sum = 0
    abs_sum = 0
    digits = 0
    for element in row:
        sum += ((pos*element))
        abs_sum += abs((pos * element))
        pos += 1
        if element != 0:
            digits += 1
    if sum != 0:
        hypothesis = False
        break
    print(str(row_num) + ":" + str(digits) + " " + str(abs_sum))
    row_num += 1
print(hypothesis)

```