Modern Complexity Theory Homework Zero

The aim of problem set is to help you to test, and if needed to brush up, up on the mathematical background needed to be successful in this course.

- **Collaboration:** You can collaborate with other students that are currently enrolled in this course in brainstorming and thinking through approaches to solutions but you should write the solutions on your own and cannot share them with other students.
- **Owning your solution:** Always make sure that you "own" your solutions to this other problem sets. That is, you should always first grapple with the problems on your own, and even if you participate in brainstorming sessions, make sure that you completely understand the ideas and details underlying the solution. This is in your interest as it ensures you have a solid understanding of the course material, and will help in the midterms and final. Getting 80% of the problem set questions right on your own will be much better to both your understanding than getting 100% of the questions through gathering hints from others without true understanding.
- **Serious violations:** Sharing questions or solutions with anyone outside this course, including posting on outside websites, is a violation of the honor code policy. Collaborating with anyone except students currently taking this course or using material from past years from this or other courses is a violation of the honor code policy.
- **Submission Format:** The submitted PDF should be typed and in the same format and pagination as ours. Please include the text of the problems and write **Solution X:** before your solution. Please mark in gradescope the pages where the solution to each question appears. Points will be deducted if you submit in a different format.

By writing my name here I affirm that I am aware of all policies and abided by them while working on this problem set:

Your name: Kunwar Shaanjeet Singh Grover, Email: kunwar.shaanjeet@students.iiit.ac.in

Collaborators: Zeeshan Ahmed, Alapan Chaudhuri

Questions

Please solve the following problems. Some of these might be harder than the others, so don't despair if they require more time to think or you can't do them all. Just do your best. Also, you should only attempt the bonus questions if you have the time to do so. If you don't have a proof for a certain statement, be upfront about it. You can always explain clearly what you are able to prove and the point at which you were stuck. Also, for a non bonus question, you can always simply write **"I don't know"** and you will get 15 percent of the credit for this problem.

The discussion board for this course will be active even before the course starts. If you are stuck on this problem set, you can use this discussion board to send a private message to all instructors under the `e-office-hours` folder.

This problem set has a total of **50 points** and **11 bonus points**. A grade of 50 or more on this problem set is considered a perfect grade. If you get stuck in any questions, you might find the resources in the CS 121 background page at `https://cs121.boazbarak.org/background/` helpful.

Problem 0 (5 points): Read fully the **Mathematical Background Chapter** from the textbook at `https://introtcs.org/public/lec_00_math_background.pdf`. This is probably the most important exercise in this problem set!!

Solution 0: I certify that I fully read the mathematical background chapter

0.0.1 Logical operations, sets, and functions

These questions assume familiarity with strings, functions, relations, sets, and logical operators. We use an indexing from zero convention, and so given a length n binary string $x \in \{0,1\}^n$, we denote coordinates of x by $x_0, \dots, x_{n-1}$. We use $[n]$ to denote the set $\{0, 1, \dots, n-1\}$.

Question 1 (3 points): Write a logical expression $\varphi(x)$ involving the variables x_0, x_1, x_2 and the operators $\wedge$ (AND), $\vee$ (OR), and $\neg$ (NOT), such that $\varphi(x)$ is true if and only if the majority of the inputs are *False*.

Solution 1: $\varphi(x)$:

$$(\neg x_0 \wedge \neg x_1 \wedge x_2) \vee (\neg x_0 \wedge x_1 \wedge \neg x_2) \vee (x_0 \wedge \neg x_1 \wedge \neg x_2) \vee (\neg x_0 \wedge \neg x_1 \wedge \neg x_2)$$

Question 2: Use the logical quantifiers $\forall$ (for all), $\exists$ (exists), as well as $\wedge, \vee, \neg$ and the arithmetic operations $+, \times, =, >, <$ to write the following:

Question 2.1 (3 points): An expression $\psi(n,k)$ such that for every natural numbers n, k, $\psi(n,k)$ is true if and only if k divides n.

Solution 2.1: $\psi(n,k)$:

$$\exists_{c \in \mathbb{N}} (n = ck)$$

Question 2.2 (3 points bonus): An expression $\varphi(n)$ such that for every natural number n, $\varphi(n)$ is true if and only if n is a power of three.

Solution 2.2: $\varphi(n)$:

$$(\forall_{i>0, i\in\mathbb{N}}\ a_i = 3a_{i-1}) \wedge (a_0 = 1) \wedge (\exists_{a_i}\ n = a_i)$$

Question 3: In this question, you need to describe in words sets that are defined using a formula with quantifiers. For example, the set $S = \{x \in \mathbb{N} : \exists_{y\in\mathbb{N}} x = 2y\}$ is the set of even numbers.

Question 3.1 (3 points): Describe in words the following set S:

$$S = \{x \in \{0,1\}^{100} : \forall_{i\in\{0,\ldots,98\}} x_i = x_{i+1}\}$$

(Recall that, as written in the mathematical background chapter, we use zero-based indexing in this course, and so a string $x \in \{0,1\}^{100}$ is indexed as $x_0 x_1 \cdots x_{99}$.)

Solution 3.1: S is the set of binary strings of length 100 with all bits as either 0 or 1.

Question 3.2 (3 points): Describe in words the following set T:

$$T = \{x \in \{0,1\}^* : |x| > 1 \text{ and } \forall_{i\in\{2,\ldots,|x|-1\}} \forall_{j\in\{2,\ldots,|x|-1\}} i \cdot j \neq |x|\}$$

Solution 3.2: T is the set of all binary strings of prime length.

Question 4: This question deals with sets, their cardinalities, and one to one and onto functions. You can cite results connecting these notions from the course's textbook, MIT's "Mathematics for Computer Science" or any other discrete mathematics textbook.

Question 4.1 (4 points): Define $S = \{0,1\}^6$ and T as the set $\{n \in [100] \mid n \text{ is prime }\}$. *Prove or disprove:* There is a one to one function from S to T.

Solution 4.1:

$$\begin{aligned} |S| &= 2^6 = 64 \\ |T| &= 25 \\ \therefore |S| &> |T| \end{aligned}$$

The problem of mapping elements from S to T can be seen as placing $|S|$ elements in $|T|$ bins. Since $|S| > |T|$ at least one bin will be mapped to atleast 2 elements. Thus, for any function $f : S \to T$, there is atleast one element in T which is mapped to atleast 2 elements in S. Hence, no one-to-one function can exist from S to T.

Question 4.2 (4 points): Let $n = 100$, $S = [n] \times [n] \times [n]$ and $T = \{0,1\}^n$. *Prove or disprove:* There is an onto function from T to S.

Solution 4.2:

$$\begin{aligned} |S| &= n^3 = 10^6 \\ |T| &= 2^n = 2^{100} \\ \therefore |S| &\leq |T| \end{aligned}$$

Lemma 1. *Given two sets S and T such that, $|T| \leq |S|$, There exists an onto function from S to T.*

Proof: Let $S' \subseteq S \mid |S'| = |T|$.

Map $s'_1 \to t_1, s'_2 \to t_2, \ldots$ where $s'_i \in S'$ and $t_i \in T$. This is a bijective function as every element in S' is mapped uniquely and every element of T is mapped, since the size of sets is same.

Map each element in $S \backslash S'$ to any element in T. This way, we map S to T, such that the map is onto. Since there exists such a map, there exists an onto function from S to T.

□

By Lemma 1, since $|S| \leq |T|$, there exists an onto function from T to S.

Question 4.3 (4 points): Let $n = 100$, let $S = \{0,1\}^{n^3}$ and T be the set of all functions mapping $\{0,1\}^n$ to $\{0,1\}$. *Prove or disprove:* There is a one to one function from S to T.

Solution 4.3:

$$|S| = 2^{n^3} = 2^{10^6}$$
$$|T| = 2^{2^{100}}$$
$$|S| \leq |T|$$

Lemma 2. *Given two sets S and T such that, $|S| \leq |T|$, There exists a one-to-one function from S to T.*

Proof: Let $T' \subseteq T \mid |T'| = |S|$.

Map $s_1 \to t'_1, s_2 \to t'_2, \ldots$ where $s_i \in S$ and $t'_i \in T'$. This is a bijective function as every element in S is mapped uniquely and every element of T' is mapped, since the size of sets is same.

Since every element of S is uniquely mapped to an element in $T' \subseteq T$ such that the map is one-to-one we can extend this to a one-to-one function $f : S \to T$. Hence, there exists a function from S to T such that it is one-to-one.

□

Since $|S| \leq |T|$, using Lemma 2, there exists a one-to-one function from S to T.

Question 5.1 (5 points): Prove that for every finite sets A, B, C, $|A \cup B \cup C| \leq |A| + |B| + |C|$.

Solution 5.1: To prove this inequality, we use Inclusion/Exclusion Principle

$$|A \cup B| \leq |A| + |B|$$

Using the the fact that $|(A \cup B) \cap C| \geq 0$,

$$|A \cup B| - |(A \cup B) \cap C| \leq |A \cup B|$$

Using the above two inequalties,

$$|A \cup B| - |(A \cup B) \cap C| \leq |A| + |B|$$

Adding $|C|$ on both sides,

$$|A \cup B| + |C| - |(A \cup B) \cap C| \leq |A| + |B| + |C|$$

Using the Inclusion/Exclusion Principle $|A \cup B| = |A| + |B| + |A \cap B|$,

$$|A \cup B \cup C| \leq |A| + |B| + |C|$$

Hence proved.

Question 5.2 (5 points bonus): Prove that for every finite sets A, B, C, $|A \cup B \cup C| \geq |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C|$.

Solution 5.2:

Using Inclusion/Exclusion Principle,

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C|$$

Since, $|A \cap B \cap C| \geq 0$,

$$|A \cup B \cup C| \geq |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C|$$

Hence Proved.

0.0.2 Graphs

Thee following two questions assume familiarity with basic graph theory. If you need to look up or review any terms, the CS 121 background page at `https://cs121.boazbarak.org/background/` contains several freely available online resources on graph theory. This material also appears in Chapters 13,14,16 and 17 of the CS 20 textbook "Essential Discrete Mathematics for Computer Science" by Harry Lewis and Rachel Zax.

Question 6.1 (5 points): Prove that if G is a directed acyclic graph (DAG) on n vertices, if u and v are two vertices of G such that there is a directed path of length $n-1$ from u to v then u has no in-neighbors.[1]

Solution 6.1:

Lets assume $\exists_{u' \in V}\ u' \mid (u', u) \in E$, where V is the set of all vertices in G and E is the set of all edges in G.

[1] *Hint:* You can use the topological sorting theorem shown in the mathematical background chapter.

Since there is a path $\{u', u\}$ of length 1 and a path $\{u, u_0, u_1, \ldots, v\}$ for $u_i \in V$ of length $n-1$, there exists a path $P = \{u', u, u_0, u_1, \ldots, v\}$ of length n.

In a path of length n, there are $n+1$ vertices. Since there are $n+1$ vertices in a set and only n total vertices, by pigeonhole principle, atleast 1 vertex is repeated.

Since, at least 1 vertex is repeated, there exists a cycle in the path. But by defination of DAG, there cannot exists a cycle in the graph. This is a contradiction and hence the assumption that $\exists u' \in V$ is wrong.

Therefore, indegree of u is zero. Hence, proved.

Question 6.2 (5 points): Prove that for every undirected graph G of 1000 vertices, if every vertex has degree at most 4, then there exists a subset S of at least 200 vertices such that no two vertices in S are neighbors of one another.

Solution 6.2:

Let V denote set of all vertices in G and U denote set of all edges in G.

Let use use the following algorithm to obtain S:

Algorithm 1 Algorithm to obtain S

while $\exists u \in V$ **do**
  $S \leftarrow S \bigcup \{u\}$
  $V \leftarrow V - \{u\}$
  $V \leftarrow V - \{v \mid (u, v) \in E\}$
  $E \leftarrow E - \{(u, v) \mid (u, v) \in E\}$
end while

Claim 3. *Algorithm 1 produces a set S such that no two vertices have an edge between them in G and is of size atleast $\frac{|V|}{5}$.*

Proof: In any iteration when we choose a vertex, since we delete all vertices having an edge to the chosen vertex, Any chosen vertex in further iterations cannot have an edge to a vertex in S. Thus the chosen set S has no two vertices with an edge between them in G.

Since, maximum degree of any node is 4, the maximum number of neighbours it can have is 4. In the worst case, in each iteration, the chosen vertex is removed from V and at max 4 neighbours are removed. Thus, in each iteration, at max 5 vertices are removed.

Number of iterations $\geq \frac{V}{5}$.

In each iteration we add 1 element to S. Therefore $|S| =$ Number of iterations $\geq \frac{|V|}{5}$.

$\square$

Using Claim 3,

$$
\begin{aligned}
|S| &\geq \frac{|V|}{5} \\
|S| &\geq \frac{1000}{5} \\
|S| &\geq 200
\end{aligned}
$$

Hence proved.

0.0.3 Big-O Notation

Question 7: For each pair of functions f, g below, state whether or not $f = O(g)$ and whether or not $g = O(f)$.

Question 7.1 (3 points): $f(n) = n(\log n)^3$ and $g(n) = n^2$.

Solution 7.1:

$$f = O(g) : \limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty$$

$f = O(g)$:

$$
\begin{aligned}
\frac{f(n)}{g(n)} &= \frac{n(\log n)^3}{n^2} \\
&= \frac{(\log n)^2}{n} \\
\limsup_{n\to\infty} \frac{f(n)}{g(n)} &= \limsup_{n\to\infty} \frac{(\log n)^2}{n} \\
&= 0 \\
&< \infty
\end{aligned}
$$

Hence, $f = O(g)$ holds

$g = O(f)$

$$
\begin{aligned}
\limsup_{n\to\infty} \frac{g(n)}{f(n)} &= \limsup_{n\to\infty} \frac{n}{(\log n)^2} \\
&\to \infty
\end{aligned}
$$

Hence, $g = O(f)$ does not hold.

Question 7.2 (3 points): $f(n) = n^{\log n}$ and $g(n) = n^2$.

Solution 7.2:

$f = O(g)$:

$$\begin{aligned}
\limsup_{n\to\infty} \frac{f(n)}{g(n)} &= \limsup_{n\to\infty} \frac{n^{\log n}}{n^2} \\
&= \frac{n^{\log n - 2}}{} \\
&\to \infty
\end{aligned}$$

Hence, $f = O(g)$ does not hold.

$g = O(f)$:

$$\begin{aligned}
\limsup_{n\to\infty} \frac{g(n)}{f(n)} &= \limsup_{n\to\infty} n^{2-\log n} \\
&= 0 \\
&< \infty
\end{aligned}$$

Hence, $g = O(f)$ holds.

Question 7.3 (3 points bonus): $f(n) = \binom{n}{\lceil 0.2n \rceil}$ (where $\binom{n}{k}$ is the number of k-sized subsets of a set of size n) and $g(n) = 2^{0.1n}$.[2]

Solution 7.3:

[2] *Hint:* one way to do this is to use Stirling's approximation (`https://en.wikipedia.org/wiki/Stirling%27s_approximation`).