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: Rishabh Singhal, rishabh.singhal@research.iiit.ac.in

Roll number: 20171213

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 $x_0, \ldots, x_{n-1}$. We use $[n]$ to denote the set $\{0, 1, \ldots, n-1\}$.

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: $\varphi(x) = (\neg x_0 \wedge \neg x_1) \vee (\neg x_1 \wedge \neg x_2) \vee (\neg x_2 \wedge \neg x_0)$

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: $\forall\ n, k \in \mathbb{N},\ \psi(n,k) = (k|n)$

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: $\forall\ n \in \mathbb{N},\ \ \phi(n) = (\exists_{k \in \mathbb{N}}\ n = 3^k)$

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: It is a set of 2 elements (strings) i.e. $\{0\}^{100}$ or 000..0 (100 *times*) and $\{1\}^{100}$ or 111..1 (100 *times*).

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: This set contains all the binary strings $x \in \{0,1\}^{len}$ having length as a prime number, i.e. $len = 2, 3, 5, ..., p, ...$

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: T contains primes < 100, which are 25 in number, therefore $|T| = 25$. But, the number of elements in S is $2^6 = 64$, therefore $|S| = 64$. Clearly, $|S| > |T|$. Therefore, there **can not exist a one-to-one function** from S to T, as more than 1 elements of S should be mapped with one element of 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: The number of elements in S will be 100^3, i.e. $|S| = 10^6$. And, the number of elements in T is 2^n, i.e. $|T| = 2^{100}$ as $n = 100$. Clearly, $2^{100} \sim 10^{30} > 10^6$, hence $|T| > |S|$. Hence, there **must exist an onto function** from T to S, where each element in S is an image of some element in T, as clearly number of elements in T is more than 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: The number of elements in S is $2^{n^3} = 2^{100^3} = 2^{10^6}$, i.e. $|S| = 2^{10^6}$. While the number of possible functions from $\{0,1\}^n$ to $\{0,1\}$ is 2^{2^n} as there are 2 choices from $\{0,1\}$ for every 2^n string or (ordered tuple) in $\{0,1\}^n$, hence $|T| = 2^{2^n} = 2^{2^{100}}$. Therefore, $|T| > |S|$ and hence there **can exist a one-to-one function** from S to T, mapping lesser number of elements to unique ($|S|$) elements in 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: As, $|A \cup B| = |A| + |B| - |A \cap B|$ and $|A \cap B| \geq 0$,

$$|A \cup B| \leq |A| + |B| \tag{1}$$

$$
\begin{aligned}
\text{now, } |A \cup B \cup C| &= |(A \cup B) \cup C| \\
&\le |(A \cup B)| + |C| \text{ ...using (1)} \\
&\le |A| + |B| + |C| \text{ ...using (1) again}
\end{aligned}
$$

$$\text{Therefore, } |A \cup B \cup C| \le |A| + |B| + |C|$$

Hence Proved.

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

Solution 5.2: Let $|A \cup B| = |A| + |B| - |A \cap B|$ be equation (1),

$$
\begin{aligned}
|A \cup B \cup C| &= |(A \cup B) \cup C| \\
&= |A \cup B| + |C| - |(A \cup B) \cap C| \text{ ...using (1)} \\
&= |A| + |B| - |A \cap B| + |C| - |(A \cap C) \cup (B \cap C)| \text{ ...using (1) again and distributive law} \\
&= |A| + |B| + |C| - |A \cap B| - (|A \cap C| + |B \cap C| - |(A \cap C) \cap (B \cap C)|) \text{ ...using (1) again} \\
&= |A| + |B| + |C| - |A \cap B| - |A \cap B| - |A \cap B| + |A \cap B \cap C|
\end{aligned}
$$

Now, as $|A \cap B \cap C| \ge 0$. Therefore,

$$|A \cup B \cup C| \ge |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| \tag{2}$$

Hence Proved.

0.0.2 Graphs

The 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: Given: A DAG G which have a path from u to v for length $n-1$. Let this path be $p = (u, a_2, a_3, ..., a_k, v)$.
Claim: There is no repetition of vertices in this path i.e. all the vertices in this path must be unique.

Proof. Let's assume there is a repetition of vertex, then there must exist a path from that vertex (from the first occurrence in the path) to the same vertex (next occurrence, which is the repetition). And hence there must exist a cycle, which leads to a contradiction as the given graph is a **DAG**.

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

Hence, all the vertices in the given path must be unique.

And as the path length is $n-1$, the path must consists of all the n vertices. And hence, it can be seen that all the vertices other u are on a path from u to v. Therefore, there can't be any back edge from any node to u due to acyclic property. Hence proved. $\square$

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's consider the vertices one by one, and construct the subset S (by construction). Consider following Algorithm:

1. Consider an unmarked vertex randomly and add that to the subset S, and mark the vertices which are directly connected this vertex (there can be only 4 such vertices, as the degree of each vertex is at most 4).
2. Repeat Step 1, until no such unmarked vertex is left.

Now, as all the neighbours of each vertices are not taken in set S, it is evident that all the vertices in S must be unconnected or there should be no edge between any two vertices of the set S (so it follows the required property).
Next, as it can be seen for every vertex ≤ 4 vertices are marked (or surely not included in the set S) therefore let's say there are n vertices in the set, then $\leq 4n$ vertices would have not been included in the set.
As total number of vertices is 1000, therefore $1000 \leq n + 4n$, which gives $200 \leq n$. Hence at least 200 vertices will be there in such set S.

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: Here, $f(1) = 0$ and $g(1) = 1$ therefore, $f(1) < g(1)$. Let's see if $g(n) > f(n)$ in general, let $u(n) = (\log n)^3$ and $v(n) = n^2$. If $u(n) < v(n)$ for $\forall n \geq 0$ then surely, $g(n) > f(n)$. So let's see if $u(n) < v(n)$. Now, differentiate both,

$$u'(n) = 3\frac{(\log n)^2}{n}$$
$$v'(n) = 1$$

Now, $u'(n)$ will always be $< 1 \ \forall \ n \geq 1$ therefore, $v'(n) > u'(n)$ and $v(0) > u(0)$. Hence $v(n) > u(n) \ \forall \ n \geq 0$. And hence multiplying by n on both sides.

$$g(n) > f(n) \ \forall \ n \geq 0 \tag{3}$$

Therefore, $f = O(g)$.

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

Solution 7.2: Let's see if there exists some $n > N_0$ where $n, N_0 \in \mathbb{N}$ such that $a * f(n) > g(n)$. Let,

$$\begin{aligned} f(n) &> g(n) \\ n^{\log n} &> n^2 \\ \log n &> 2 \\ n &> 2^2 \text{ assuming base of log is 2} \end{aligned}$$

Hence, for $n > 4$, $f(n) > g(n)$ therefore, $g = O(f)$.

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: Using sterling approximation,

$$n! \sim \sqrt{2\pi n}(\frac{n}{e})^n \tag{4}$$

$$\begin{aligned} \binom{n}{\lceil 0.2n \rceil} &= \frac{n!}{(n - \lceil 0.2n \rceil)\lceil 0.2n \rceil} \\ &\sim \frac{\sqrt{2\pi n}(\frac{n}{e})^n}{\sqrt{2\pi(n - \lceil 0.2n \rceil)}(\frac{(n-\lceil 0.2n \rceil)}{e})^{(n-\lceil 0.2n \rceil)} * \sqrt{2\pi \lceil 0.2n \rceil}(\frac{\lceil 0.2n \rceil}{e})^{\lceil 0.2n \rceil}} \\ &\sim \frac{\sqrt{n} * n^n}{\sqrt{2\pi(n - \lceil 0.2n \rceil)\lceil 0.2n \rceil} * (0.8n)^{0.8n} * (0.2n)^{0.2n}} \\ &\sim \frac{a * n^n}{\sqrt{n} * (\frac{4}{5}n)^{\frac{4}{5}n} * (\frac{1}{5}n)^{\frac{1}{5}n}} \\ &\sim \frac{a * 5^n}{\sqrt{n} * 4^{\frac{4}{5}n}} \geq \frac{a * 5^{0.2n}}{\sqrt{n}} \end{aligned}$$

Where a is a constant. Now, let's see if $a * f \geq g$ holds i.e.

$$\frac{a * 5^{0.2n}}{\sqrt{n}} \geq 2^{0.1n} \tag{5}$$

if above equation is correct,

$$a * \frac{5^{0.2n}}{2^{0.1n}} \geq \sqrt{n} \tag{6}$$

Which surely holds, as on one side it's square-root function while on other side it is exponential function. Hence, $g = O(f)$.

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