CS 121 Homework Zero: Fall 2021

The aim of problem set is to help you to test and, if needed, to brush up on the mathematical background needed to be successful in CS 121. This problem set will be due in the first week of class, but you are encouraged to start working on it over the summer.

Some policies: (See the course syllabus at

`http://madhu.seas.harvard.edu/courses/Fall2021/syllabus.html` for the full policies.)

- **Collaboration:** You can collaborate with other students that are currently enrolled in this course (or, in the case of homework zero, planning to enroll in this course) in brainstorming and thinking through approaches to solutions but you should write the solutions on your own: you must wait one hour after any collaboration or use of notes from collaboration before any writing in your own solutions to submit.
- **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.
- **Late Day Policy:** To give students some flexibility to manage your schedule, you are allowed a net total of **eight** late days through the semester, but you may not take more than **two** late days on any single problem set.

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: Jamin Liu 61463690

Collaborators: Brad Campbell

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** (Chapter 1) from the textbook at `http://madhu.seas.harvard.edu/courses/Fall2021/book.pdf`. This is probably the most important exercise in this problem set!!

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

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

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: $\psi(n,k) = \forall_{n \in \mathbb{N}} \forall_{k \in \mathbb{N}} \exists_{i \in \mathbb{N}} i \times k = n$

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. (Note that the problem is much harder if you replace "three" by "ten"; if you think you have a solution that would work for that problem, check that you haven't used a banned operation like exponentiation or used a variable out of scope (e.g. after $(\forall i \exists f(i)) \wedge \forall j \exists g(j)$, you can't use $f(i)$ or $g(j+1)$).)

Solution 2.2:

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:

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

(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 100-length binary strings whose digits are all the same.

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

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

Solution 3.2: T is the set of all binary strings whose cardinalities are prime.

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: The cardinality of S can be computed as $2^6 = 64$ while the cardinality of T is 25. Let $s_0 ... s_{63} \in S$ and $t_0 ... t_{24} \in T$. Let F be a function such that $F(s_i) = t_i$ for all $i \in [25]$. For all s_j where $j > 24$, F cannot map s_j to a $t_n \in T$ such that $F(s_i) \neq t_n$ where $i \in [25]$. As such, there cannot exist a one to one function 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: The cardinality of T can be described as 2^{100} while the cardinality of S can be described as 100^3. As such $|S| < |T|$. Let $t_0 ... t_{|S|-1} ... t_{|T|-1} \in T$ and $s_0 ... s_{|S|-1} \in S$. Let F be a function such that $F(t_i) = s_i$ for all $i \in [|S|]$. As such, for all $s \in S$ there exists a $t \in T$ such that $F(t) = s$, so 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: The cardinality of S can be computed as 2^{100^3} while the cardinality of T is $2^{2^{100}}$. As such $|S| < |T|$. Let $s_0 ... s_{|S|} \in S$ and $t_0 ... t_{|S|} ... t_{|T|} \in T$. Let F be a function such that $F(s_i) = t_i$ for all $i \in [|S|]$. Then for all s_i in S, there exists a unique t_i such that $F(s_i) = t_i$. As such, there exists 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: Let us assume for the sake of contradiction, that $|A \cup B \cup C| > |A| + |B| + |C|$. Then, the cardinality of $A \cup B \cup C$ would have to be greater than the combined cardinalities of A , B, and C. This would imply that $A \cup B \cup C$ contains elements that are not contained in A , B, or C, which is contradictory as $A \cup B \cup C$ can solely be composed of elements in A , B, or C by definition of union. $|A \cup B \cup C|$ can at most be equal to $|A|+|B|+|C|$ if A , B, and C share no common elements. Otherwise, if there exists intersection between A , B, and C, these will be counted multiple times by $|A|+|B|+|C|$ but only once by $|A \cup B \cup C|$, leading $|A \cup B \cup C| < |A|+|B|+|C|$.

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:

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: Let G be a directed acyclic graph on n vertices and let $f : V \to \mathbb{N}$ be a layering of G. Let u and v be vertices of G such that there is a directed path of length $n-1$ from u to v. Hence there exists the path $s_0 ... s_{n-1}$ containing all vertices of G where $s_0 = u$ and $s_{n-1} = v$ and for every $i \in [n]$ the edge $s_i \to s_{i+1}$ is present in G. As such, since f is a layering of G, $f(u) < f(s_1) ... < f(n-1)$. Suppose, towards contradiction, that there exists an in-neighbor of u, s_x in G where $x \in [n]$. Then, $f(s_x) < f(u)$. However, this is not possible as the value of $f(u)$ is the lowest for all vertices of G. Thus, u has no in-neighbors.

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 be the set of all vertices in G such that no two vertices are neighbors of one another. For every vertex $s \in S$, there exist at most 5 vertices in V, the set of all vertices G, which cannot belong to S, s itself, and up to 4 of its neighbors. As such, the cardinality of S is at least one fifth of the cardinality of V, so S must have at least 200 vertices.

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 $g(n) = n^2$.

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

Solution 7.1: $f = O(g)$ and $g \neq O(f)$

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

Solution 7.2: $f \neq O(g)$ and $g = O(f)$

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:

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