

Week 2: Counting Past Infinity

The problems are designed to develop your skills with formal set cardinalities and how they relate to functions, provide some practice with inductive reasoning, and get you acquainted with your cohort-mates.

Collaboration: You should work on the problems yourself, before discussing with others, and with your cohorts are your cohort meeting. By the Assessed Cohort Meeting, you and all of your cohortmates, should be prepared to present and discuss solutions to all of the assigned problems. In addition to discussing with your cohortmates, you may discuss the problems with any other current CS3102 students you want, and use any resources you want except for any materials from previous offerings of this course or complete solutions that might be available on the web, which are not permitted. **At the end of your assessed cohort meeting, your Cohort Coach will assign one of these problems as a writeup. You may not collaborate on this writeup with your cohort-mates, but you may use notes taken before or during your assessed cohort meeting.**

Problem 1: Cohortommentations

Reflect back on your experience from last week. Consider, but do not share your answers to, these questions:

1. Did you employ any of the video watching tips? Did you find any of them helpful? Did any have a negative impact on your learning? Are there any you think might help if you employed them in the future?
2. How prepared were you for your first cohort meeting? Should you have done more to prepare in advance?
3. Did you find your interactions with your cohort mates to be fairly natural? How can you personally help to improve your group dynamic for future semesters?

Next, share and discuss these questions with your cohorts:

1. What is working well and what is not working for the course so far?
2. What are things the course staff can do to make things better?
3. What are things you or your cohort can do to improve your ability to interact and learn that you think might be helpful for other cohorts to know about?

Once you've discussed these with your cohort, have one person share at least one of your answers to question 3 on the #week2 discord channel. You should read the other cohorts answers there also, of course, and if you can, connect your suggestion to other cohorts' answers.

Problem 2: Infinite Dominoes

A domino is a tile with an unordered pair of numbers on it (e.g. 0, 5 or 3, 3). Dominoes come in sets containing all pairs of natural numbers less than or equal to some upper bound.

A pack of "double 6" dominoes will contain all unordered pairs of values from the set $\{0, 1, 2, 3, 4, 5, 6\}$ (there will be $28 = 7 + 6 + 5 + 4 + 3 + 2 + 1$ total). A pack of "double 3" dominoes will contain all unordered pairs of values from the set $\{0, 1, 2, 3\}$ (there will be 10 total).

A *domino chain* is a sequence of dominoes, with no duplicates, ordered so that the second value of each domino matches the first value of the next. The domino sequence $(1, 2)(2, 5)(5, 5)(5, 0)$ is a valid domino chain, whereas $(1, 2)(2, 5)(5, 5)(0, 0)$ is not.

Consider a pack of "double \mathbb{N} " dominoes, which contains all of the infinitely-many unordered pairs of natural numbers. Show that there is an uncountable number of infinite-length domino chains that can be constructed from a pack of "double \mathbb{N} " dominoes.

Problem 3: Cantor's Proof

The [Uncountability](#) video presents a proof by contradiction that for any set S , $|2^S| > |S|$. While this may seem obvious for finite sets, Cantor's proof shocked the math world because it also applied to infinite sets, demonstrating for the first time that infinite sets can have different cardinalities.

For this problem, we will be checking that you understand the proof. Be prepared to answer questions like “What statements from the proof are contradictory?”, “How is the set P defined?”, “Why didn't this proof require the sets to be finite?”

Problem 4: **NOR equals AON** (based on Exercise 3.8 in TCS book)

Let $NOR : \{0, 1\}^2 \rightarrow \{0, 1\}$ defined as $NOR(a, b) = NOT(OR(a, b))$. Prove that $\{NOR\}$ is equivalent to AND , OR , NOT . In other words, show that any function that can be computed by AND, OR, NOT can also be computed using just NOR , and vice-versa.

Problem 5: **MAJ, NOT, I equals AON** (based on Exercise 3.6 in TCS book)

Let $MAJ : \{0, 1\}^3 \rightarrow \{0, 1\}$ and $NOT : \{0, 1\} \rightarrow \{0, 1\}$ be defined as in lecture. Let $I : \{0, 1\}^0 \rightarrow \{0, 1\}$ simply give the constant value 1. Prove that $\{MAJ, NOT, I\}$ is equivalent to AND, OR, NOT . In other words, show that any function that can be computed by AND, OR, NOT can also be computed using combinations of $\{MAJ, NOT, I\}$, and vice-versa.

Problem 6: **Straightline Programming**

For this problem, you will be asked to implement several straightline programs in Python. You must adhere to the rules of straightline programs. In particular, make sure:

- Variables contain only bits.
- Function parameters are only the characters ‘0’ and ‘1’.
- Each line may only have a single function and a variable assignment (no nested functions, no conditionals, no loops).
- Variable names may not be reused within the same function.

We provide test cases for all functions. If you followed the above rules, and you pass all tests, you can be confident that your implementation is correct.

To complete this problem, download the [straightline.py](#) program. The comments in that file guide you through the assignment. Everything you must implement is marked with a comment containing “TODO”.

Problem 7: **Nate's Omission**

In the [Uncountability](#) video, Professor Brunelle accidentally left off an important part of his proof that $|\{0, 1\}^\infty| = |\{\{0, 1\}^* \rightarrow \{0, 1\}\}|$. From the justification provided in the video, you cannot conclude that the two sets are of equal size, only that one set was larger than the other.

Which set can we conclude is larger by Professor Brunelle's justification?

Provide the missing piece to show that they are the same size.