Problem Set 1

This problem set is due **at 11:59 PM** on **Thursday, September 30, 2021.**

Please make note of the following instructions:

- This assignment, like later assignments, consists of *exercises* and *problems*. **Hand in solutions to the problems only.** However, we strongly advise that you work out the exercises for yourself, since they will help you learn the course material. You are responsible for the material they cover.
- Remember that the problem set must be submitted on Gradescope. If you haven't done so already, please signup for QRMT FC-0309-1 (FCP) Foundations of Computer Programming on Gradescope, with the entry code 4PZV7W, to submit this assignment.
- We require that your solutions are submitted as a single PDF file. Each submitted solution should start with your name, the course number, the problem number, the date, and the names of any students with whom you collaborated. Each problem should start on a new page (subparts need not follow this rule).
- You may be called upon to "give an algorithm" to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of your essay should provide the following:
  1. A description of the algorithm in English, step by step.
  2. Optionally, a proof (or indication) of the correctness of the algorithm. ("Why is this correct?" – we will explicitly ask for this on occasion). You may find it useful to include a worked example or diagram to show more precisely how and why your algorithm works.
  3. An analysis of the asymptotic running time behavior of the algorithm. (You'll learn what this is later in the semester.)

EXERCISES (NOT TO BE TURNED IN)

Art and Craft

- Learn about Rene Magritte and M.C. Escher. You really should know more about these people.
- Find some instructions/videos about constructing mechanical computers, especially "simple" ones such as a marble adding machine.
- Recursive music: check out these three pieces
  - `https://www.youtube.com/watch?v=-rfezNHtwhg`
  - `https://www.youtube.com/watch?v=mq0z-sxjNlo` (this may be rather painful and not useful unless you know a bit about music)
  - `https://www.youtube.com/watch?v=hEJy-PLyL1Q`
- Dance like this: `https://www.youtube.com/watch?v=3K3MMtoG8rY`
- Also, critically important: `https://www.youtube.com/watch?v=dQw4w9WgXcQ`

Reading

In general, you should read everything that is posted on Google Classroom. You should be able to find everything listed below for free on the internet.

- As We May Think (Vannevar Bush)
  This is on Classroom. If you don't know who this is, read up on him!
- There's Plenty of Room at the Bottom (Richard Feynman)
- Where is Science Going? (Max Planck)
  This is a full book; I just put it in here because it's interesting.

Note for the PSet: Problems 1-1 and 1-3 have been ”borrowed” from the various PSets or quizzes of FCP Monsoon’19. Please do not go out looking for answers by asking your seniors. It is important for us (and you) to know how you think, so that we can help you develop it through the duration of the course.

Problem 1-1. Evolution of Computing and Automaton in South-Asia [20 points]

In class, we skimmed over the history of computational devices across civilizations and time. Throughout this, I, with near-undetectable, Drax-like subtlety, ignored South-Asian civilizations. Your homework, should you choose to accept it, is:

(a) [12 points] List the major achievements in computing from these regions. Cite your resources - you do not need to use MLA/Chicago/APA or anything, even embedding links is fine (just make sure we can trace the source). Uncited observations should be clearly marked as speculation.

(b) [8 points] If these regions had not existed, in what ways do you think computing (and the world) would be different? Depth of thought required: Greater than one level. That is, you don’t just get to “subtract” the achievements you listed in the previous part. Think carefully about how not having certain breakthroughs and knowledge available would have changed the path of progress.

Note: Please limit your answer to a maximum of 500 words (in total, NOT for each part).

Problem 1-2. Recursion: To understand recursion, you must first understand Recursion: To under... [25 points]

Statement: Recursive problems are such that require a smaller version of the problem to be solved first. The smaller problem in turn requires an even smaller version to be solved first.

(a) [15 points] Why does this work? Can we be sure that such problems will ever halt? What do you need to do in order to ensure that they do stop at some point?

(b) [10 points] Think of an action/problem from your own life that you think can be solved recursively, and explain why it is so. List out clear steps of how you can solve the problem using the definition given in the statement.

Remember to distinguish between recursion and iteration. For your reference -

- `https://xkcd.com/244/`
- `https://bit.ly/2XrgWXA`
- `https://bit.ly/3CpO958`

Problem 1-3. More than Half-Right [20 points]

Given that you already know how to go from a real-world problem to a mathematical model, I'll state it mathematically first:

We have a list of things (say, integers). We don't know how many there are, but we are told that one of these integers repeats more than 50% of the time. How will you find that integer?

You can think of this as a huge box filled with an unknown number of coloured balls. You get to empty the box one ball at a time through a "tap" at the bottom; you cannot see how many balls are left at any point in time until the box is actually empty. You are, however, told that one of the colours comprises more than 50% of the balls. Your job is to name that colour when you finish.

Provide an algorithm to do this. How many things do you need to remember? Do you need to keep all the balls/integers or can you throw them away after you pick them? How much time would it take you, assuming it takes one unit of time to compare two balls and decide whether or not they are the same colour?

Don't overcomplicate this problem: while there is indeed an optimal solution, I do not expect any of you to get it. Just say how **you** would do it, in a step-by-step fashion, in plain English, any why you think your method would produce the correct answer.

This is one of the most important problems in streaming: when you're dealing with data (say, video or music) and you don't know when it will end, you have process it in a very different way. For example, this class of algorithms is actually running in the background every time you watch a YouTube video – of course, they're calculating things that are much more complicated than just a 50% barrier.

Problem 1-4. Boxes [20 (+ 20 extra credit) points]

We are giving you two magic boxes, creatively named "Judith Love Cohen" and " Douglas Engelbart". You can input some numbers into these boxes, and then they produce an output. Our boxes are *deterministic* – for the same inputs, they will always produce the same outputs.

Along with the boxes, we give you two "cheat sheets" – these are meant to describe what's going on inside each box, i.e., the individual mathematical steps. Unfortunately, Debayan is really bad at writing clearly, and ended up with some imprecise cheat sheets: there are *ambiguities* in certain steps! (Don't worry, we are providing an example below!)

Your task is to begin with an input number, go through the given steps to obtain an output number, and identify one ambiguity for each output. This will form an **(input, outputs, ambiguity)** set. You are given 2 points for each correct (input, outputs, ambiguity) set you provide. **Note:** You cannot use any ambiguity more than once, i.e., if you submit two (input, outputs, ambiguity) sets with the same ambiguity, only one of them will be marked.

You are part of a group of students in the class. You do not know this group (it is hidden from you), and your group will change between assignments. The **20 extra credit** points will be given to the highest scorer of your group (or shared, if multiple people have the same high score).

Example Magic Box and Ambiguity:

1. Take a number (input) that is greater than 4
2. Multiply this number by 3
3. Take the second digit in the number from the previous step and multiply the digit you took by 6
   (Notice Debayan's mistake here! He didn't say whether this was the second digit from the left or the right – this is an **ambiguity!**)
4. Print the number (output)

For the input 7, we have the following output at each step.

- 7
- 21
- 2 (second digit from left) or 1 (second digit from right)
- Output 12 or 6

Your (input, outputs, ambiguity) set:
(7, 12 OR 6, UNCLEAR IF SECOND DIGIT IS FROM RIGHT OR LEFT)

Your magic boxes are on the next page!

Magic box "Judith Love Cohen":

1. Take a number (input) that is greater than 104.
2. Increase the number by 2 until it is divisible by 7.
3. Multiply the number by its smallest divisor.
4. Add 100 to the number.
5. Multiply the number by 20 until it has at least 5 digits in it.
6. If the number is not divisible by 11, add to the number until it is.
7. Output the number

Magic Box "Douglas Engelbart":

1. Take a number (input) that is a rational number greater than 0 and less than 1
2. Multiply the number by its inverse
3. Reduce the number to its decimal form and take only the part to the left of the decimal point
4. Divide this number by 6 and repeat again.
5. Multiply the number by 1296
6. Output the number