COMS BC 3159 - S23: Problem Set 0

Introduction: Hello and welcome to COMS BC 3159 - S23! I know what you are thinking, "ugh, why do I have to do PS 0? We haven't even learned anything yet." That might be true, but we wanted to give you all an opportunity to **practice** completing and turning in assignments for this class, as after **PS 0**, we will not be so lenient and forgiving in our handling of incorrectly submitted problem sets.

This first assignment **should not** feel too arduous or time consuming – in fact, it may even seem trivial. If it is, good! It is meant to get you back in the groove of things and ready to tackle this semester, not provide too much mental stress. If, for whatever reason, you are struggling with this assignment, **PLEASE** reach out to one of us, your teaching staff – or even fellow classmates or friends. We really want to make sure everyone comes into the class confident that they can succeed.

The following exercises will mostly test some basic math concepts as well as getting you more comfortable with LaTeX. In addition to solving the problems found below, you will also need to complete the coding part of the assignment, found in the Github repo.

Finally, we'd like to remind you that all work should be yours and yours alone. This being said, in addition to being able to ask questions at office hours, you are allowed to discuss questions with fellow classmates, provided 1) you note the people with whom you collaborated, and 2) you **DO NOT** copy any answers. Please write up the solutions to all problems independently.

Without further ado, let's jump right in!

Collaborators:
AI Tool Disclosure:

Linear Algebra

Problem 1 (2 Points):

Solve the following system of linear equations for $x, y, \text{and} z$. Please show your work and use matrix algebra.

$$\begin{aligned} x + 2y + 3z &= 1 \\ 2x + 2y + 3z &= 2. \\ x + 3y + 4z &= 2 \end{aligned}$$

Hint: the inverse of: $\begin{bmatrix} 1 & 2 & 3 \\ 2 & 2 & 3 \\ 1 & 3 & 4 \end{bmatrix}$ is: $\begin{bmatrix} -1 & 1 & 0 \\ -5 & 1 & 3 \\ 4 & -1 & -2 \end{bmatrix}$

Solution 1:

Multivariable Calculus

Problem 2 (2 Points):

Suppose that we wish to minimize some loss function $\mathcal{L}$ with respect to a variable x where:

$$\mathcal{L}(x) = 3x^2 + 2x - 1$$

Given this loss function, what value of x minimizes the loss? In other words, for what value of x is this function minimal?

Solution 2:

Problem 3 (2 Points):

Find the partial derivative $\frac{\partial f}{\partial x}$ of the following function:

$$f(x, y) = 3x^2 + 2xy - 5y^2$$

Now evaluate that derivative at $(x, y) = (2, 3)$.

Solution 3:

CS Theory

Problem 4 (2 Points)

For this last problem, we will deal with the basics of asymptotic notation. For those of you who have never seen this before, do not fret! We will only really deal loosely with the notion of Big-O notation in this class. For a really simple beginner's guide, see: https://rob-bell.net/2009/06/a-beginners-guide-to-big-o-notation/.
Briefly, Big-O notation is a way for us to talk about assymptotic run-times of algorithms. For those of you who have explored sorting algorithms in past CS classes may recall that insertion sort takes $O(n^2)$ time – where n is the length of the list to sort – while merge sort takes $O(n \ln n)$ time. **What is the runtime of the following `foo` algorithm in Big-O notation? What does the following algorithm do?**

```
def sum_list(lst):
    """
    Function that returns the sum of elements in lst
    :param lst: input list
    :type  lst: [int]
    :return: sum of the list
    :rtype : int
    """
    rtn = 0
    for elt in lst:
        rtn += elt
    return rtn


def foo(lst):
    """
    Foo Algorithm
    :param lst: input list of length n
    :type  lst: [int]
    :return: ???
    :rtype : int
    """
    for elt in lst:
        if elt < 0:
            return -1
    return sum_list(lst)
```

Solution 4:

Programming

Problem 5 (2 Points)

Consider the following C code:

```
#include <stdio.h>

int main() {
    int arr[5] = {1, 2, 3, 4, 5};
    int* ptr = &arr[2];
    *ptr = 7;
    printf("%d\n", *ptr + 2);
    return 0;
}
```

What values are in the final array? What is printed out by the function?

Solution 5: