CS313E - Software Design
Assignment 4 - 100 Points

Please note the honor code policy regarding this Assignment. You may not discuss particular questions or discuss or transmit answers from the assignment with other people, except for the CS313e teaching team.

Please submit to the Gradescope!

Note: You can create a single PDF file with multiple pages and submit it to the Gradescope. You can use using any software like MS-Word or *LaTeX* to include mathematic formulas if needed. If you can not create a PDF you can write on a piece of paper, scan it or make a picture of it using your smartphone and submit multiple images into Gradescope.

You can use this guide to see how you can submit your assignment.
`https://gradescope-static-assets.s3.amazonaws.com/help/submitting_hw_guide.pdf`

You can use this video tutorial to know how to submit your grade to Gradescope
`https://www.youtube.com/watch?v=u-pK4GzpId0`.

Tasks

1. Use python to create 3 different plots of the following functions **(15 points)**:

$$f_1(n) = (2^{20}) * (n) + 2$$
$$f_2(n) = n^{(7.1)} + (2.1)^{20}$$
$$f_3(n) = 4^n - (2.1)^8$$

- Create 3 plots and limit the horizontal x-axis to $n = 5, 15, 50$. On each of the 3 plots you need to show the above 3 functions. On the first plot the x-axis is limited to 5, on second one x-axis is limited to 15 and on the 3rd one x-axis is limited to 50.
- Visualize the 3 functions in 3 colors (f_1 in red, f_2 in blue, f_3 in green).
- Describe your visualization and what you see in these 3 plots.
- Add your visualization and your python code to your PDF report file.

You can use the template implementation provided in class.

Here `https://colab.research.google.com/drive/1Th23RUUaMKbM3CHab6RgVV2RFwTT1M_9?usp=sharing`

You can run this task on google Colaboratory.

---

2. Asymptotic Notation. **(15 points)**

- Is $2^{(2n+2.3)} = O(2^n)$?
- Is $3^{(2\times n)} = O(3^n)$?

Describe your answer.

3. For each pair of functions $f(n)$ and $g(n)$, check if $f(n) = O(g(n))$?

   Functions $f(n)$ and $g(n)$ are:

   1. $f(n) = (4 \times n)^{150} + (2 \times n + 1024)^{400}$ vs. $g(n) = 20 \times n^{300} + (n + 121)^{152}$
   2. $f(n) = n^{1.4} \times 4^{2n}$ vs. $g(n) = n^{100} \times 3.99^{n}$
   3. $f(n) = 2^{log_2^n}$ vs. $g(n) = n^{1024}$

   Describe your justifications. **(30 points)**

4. Analyze the Algorithm 1 and give a Big O bound on the running time as a function of n. Carefully describe your justifications. **(20 points)**

Algorithm 1 What is the Big O of this pseudocode?

```
$i = 1$
while $i \leq n$ do
    $A[i] = i$
    i = i + 1
end while
for $j \leftarrow 1$ to $n$ do
    $i = j$
    while $i \leq n$ do
        $A[i] = i$
        $i = i + j$
    end while
end for
```

5. Analyze the Algorithm 2. What is the Big O on the running time as a function of n. Carefully describe your justifications. **(20 points)**

Algorithm 2 What is the Big O of this pseudocode?

```
$x = 0$
for $i \leftarrow 0$ to $n$ do
    for $j \leftarrow 0$ to $(i \times n)$ do
        $x = x + 10$
    end for
end for
```