Synthesis Assignment #2
CS 5800, Fall 2022
Dr. Lindsay Jamieson

DUE: October 31st, 11:59pm eastern via Canvas

Tl;dr version - create a textbook collecting your explanations of 9 different topics including problems and solutions for each. Put it in a single pdf/docx file and submit on Canvas.

Your task is to demonstrate your understanding of the material by writing a mini-textbook. For each topic we've discussed in class (up to the week before this assignment is due) you will write an explanation of the topic based on the way that you understand the concept. A complete topic list is provided at the end of the document, divided into "chapters" based on the modules as we presented them in class.

Each topic will be complemented by at least one problem and solution of your own creation (no taking problems from other sources). This means that you will (in addition to the write up) be also responsible for creating 7 problems and their solutions. NOTE: these are problems like homework problems NOT problems that you are using to explain the concept. Yes, this means that there are 9 sections that you will be creating as part of this assignment. Please clearly number and space out your sections so that I know exactly where one section ends and the other begins. If helpful, a format I would suggest is:

1 **Chapter Name** (like Asymptotic Analysis)

1.1 **Section Name** (like Big O)
<your write up goes here>
Exercise: <problem write up>
Solution: <solution write up>

1.2 **Section Name** ...

Note: these sections don't have to be terribly long or involved, but they have to demonstrate that you do understand the concepts enough to explain them and come up with exercises. There is no page limit, or page minimum, so use your best judgement. An example of the kind of thing that I'm looking for will be provided in the assignment area on Canvas.

Special topic note: The topics labeled "XXX design" are topics that are meant to distill your knowledge about the topic, and as such do not have a particular linkage to a specific thing talked about in class. These are designed to explain to the reader how you, personally, would design an algorithm for a problem using that technique.

Teamwork note: this is an assignment where it is important that you are the sole author of your work. You are allowed to discuss concepts with your classmates, but at no time are you allowed to discuss or share problems, writeups, or your overall plan for this assignment with anyone other than the professor or the TAs.

Due: October 31st, 11:59pm eastern, via Canvas
Format: a single pdf or docx file. Note that you can create this in Google docs, just please export to pdf or docx before submitting.

Topics to be covered (organizational section in bold):

Chapter 4: Graphs -

- basic definition, including mathematical notation and classes of graphs (no problem/solution required)
- graph representations (adjacency matrix and adjacency list, both weights and unweighted)
    - Original problem should be translating a weighted graph from or to an adjacency matrix and list.
- graph traversal algorithms (BFS and DFS)
    - Original problem should be a step-by-step execution of either BFS or DFS on a graph.
- connectivity and strongly connected regions
    - Original problem should be a step-by-step execution of the strongly connected regions algorithm on a graph of size > 6 vertices.

Chapter 5: Graph Algorithms -

- Dijkstra's Algorithm
    - Original problem should be a step-by-step execution of Dijkstra's algorithm on a weighted graph of size > 8 vertices.
- Bellman-Ford Algorithm
    - Original problem should be a step-by-step execution of the Bellman-Ford algorithm on a weighted graph of size > 8 vertices. It can be the graph from the previous section.
- Subset Parameters (Matchings and Domination)
    - Original problem should be to find a minimum dominating set or maximum matching on a graph of size > 10 vertices.

Chapter 6: Greedy Algorithms -

- technique definition, including specification of problems using Goddard's definition (no problem/solution required)
- minimum spanning tree and Prim's & Kruskal's Algorithm

- Original problem should be a step-by-step execution to find an MST in a weighted graph of size > 8 vertices (can be the same graph from Dijkstra/Bellman-Ford problems) using both Prim and Kruskal.

Rubric:

Topic Explanations: 30%
Example Exercises: 30%
Solutions to Exercises: 30%
Spelling and Grammar: 10%