Foundations of Algorithms, Fall 2022
Homework #2

In this course, critical thinking and problem analysis results in discovering appropriate and better, the best, algorithm for a problem; defining the algorithm in pseudocode; demonstrating (we don't usually prove) that the algorithm is correct; and determining the asymptotic runtime for the algorithm using one or more of the tools for asymptotic analysis you have studied this semester. Each algorithm you design must follow the **Pseudocode Restrictions** (Canvas page *Pseudocode Restrictions & Programming Guidelines*) when preparing your solutions to these problems.

Exercising the algorithm you design using one or more example data set(s) removes any doubt from the grader that the algorithm is structurally correct and all computations are correct. When a problem supplies data you are expected to use it to demonstrate that your algorithm is correct.

All members of the collaboration group are expected to participate fully in solving collaborative problems, and peers will assess performance at the end of the assignment. Note, however, that each student is required to write up their solutions individually. Common solution descriptions from a collaboration group will not be accepted. Furthermore, to receive credit for a collaboration problem, each student in the collaboration group must actively and substantially contribute to the collaboration. This implies that no single student should post a complete solution to any problem at the beginning of the collaboration process.

You are permitted to use Internet resources while solving problems, however complete references must be provided. Please follow the Sheridan Libraries' citation guidance at "*Citing Other Things* - HOW DO YOU CITE AN INTERVIEW, A TWEET, OR A PUBLIC WEB PAGE?" and continue to the APA Academic Writer *Sample References*. Additional example citations are provided at the Purdue University, Purdue Online Writing Lab (OWL), College of Liberal Arts *Reference List: Electronic Sources*. You can also use training provided by Victoria University Library, Melbourne Australia on *Harvard Referencing: Internet/websites*

All written assignments are expected to be typed in a word processor, the preferred submission type is PDF allowing for ease of grading. When using equations, make use of the built-in equation editor for these tools. Try to avoid scanning hand-drawn figures, if needed please ensure they are readable. Only scans from flat bed scanners or printers will be accepted; all scans from phones or phone photos will be rejected. The grader will make the determination of "readability". If the assignments are not legible, they will be returned to the student with a grade of zero.

1. [10 points] Show that if the complement of a language, L^C, is undecidable, then L itself is undecidable. *Hint*: "show" does not require a proof, whereas "prove" does require a proof.

2. [10 points] CLRS 34.3-2: Show that the $\leq_P$ relation is a transitive relation on languages. That is, show that if $L_1 \leq_P L_2$ and $L_2 \leq_P L_3$, then $L_1 \leq_P L_3$.

3. [15 points] What level of the Chomsky hierarchy do each of these languages fall into, and why? *Hint*: Consider the *Pumping Lemma - Regular*, *Context Free Grammars*, or *General Grammars* when solving these problems.

   (a) $L_1 = \{a^n b^n | 0 \leq n \leq 1000\}$

   (b) $L_2 = \{a^n b^n | n \geq 0\}$

   (c) $L_3 = \{a^n b^m | n, m \geq 0\}$

4. [10 points] Prove that $\chi(G)$ is no less than the size of any maximal clique of G. **Do not** assume G is a complete graph.

   (a) **Definition of a clique**: A complete graph, K_n, is a graph with n vertices such that every vertex is connected to every other vertex by an edge. A clique is a complete subgraph of $G = \{V, E\}$.

   (b) **Definition of maximum and maximal cliques**: See the document HW2_Q4_Illustration.pptx

   (c) **Definition of the graph coloring problem**: The chromatic number of a graph, $\chi(G)$, is the minimum number of colors (labels) required to color (label) the vertices of G such that adjacent vertices have different colors (labels).

5. [15 points] CLRS 34.1-3: Give a formal encoding of directed graphs as binary strings using an adjacency matrix representation. Do the same using an adjacency-list representation. Use the directed graph in Figure 22.2, page 590, and submit both formal encodings. Argue that the two representations are polynomially related.

(The assignment continues on next page)

The following information is provided to aid students in solving some of these homework problems. To prove a problem is NP-complete:

(a) Chapter 34.4 NP-Completeness proofs page 1073 specifies the required steps in an NP-complete proof.

(b) These steps will require you to also use Chapter 34.3 NP-completeness and reducibility pages 1063.

(c) These steps will require you to also use Chapter 34.2 Polynomial-time verification page 1058.

---

6. [20 points] Suppose you're helping to organize a summer sports camp, and the following problem comes up. The camp is supposed to have at least one counselor who's skilled at each of the n sports covered by the camp (tennis, field hockey, corn hole, and so on). They have received job applications from m potential counselors. For each of the n sports, there is some subset of the m applicants qualified in that sport. The question is "For a given number $k < m$, is is possible to hire at most k of the counselors and have at least one counselor qualified in each of the n-sports?" We'll call this the *Efficient Recruiting Problem.*

   Prove that Efficient Recruiting is NP-complete.

7. [20 points] ***Collaborative Problem*** : We start by defining the *Independent Set Problem (IS)*. Given a graph $G = (V, E)$, we say a set of nodes $S \subseteq V$ is *independent* if no two nodes in S are joined by an edge. The Independent Set Problem, which we denote IS, is the following. Given G, find an independent set that is as large as possible. Stated as a decision problem, IS answers the question: "Does there exist a set $S \subseteq V$ such that $|S| \geq k$?" Then set k as large as possible. For this problem, you may take as given that IS is NP-complete.

   A store trying to analyze the behavior of its customers will often maintain a table, A, where the rows of the table correspond to the customers and the columns (or fields) correspond to products the store sells. The entry $A[i, j]$ specifies the quantity of product j that has been purchased by customer i. For example, Table 1 shows one such table.

   Table 1: Example Customer Tracking Table

Customer	Pencils	Chalk	Soda	Printer Paper
David	0	6	0	3
Doja	2	3	0	0
Olivia	0	0	4	7
Bruno	0	3	1	5

   One thing that a store might want to do with this data is the following. Let's say that a subset S of the customers is *diverse* if no two of the customers in S have ever bought the same product (i.e., for each product, at most one of the customers in S has ever bought it). A diverse set of customers can be useful, for example, as a target pool for market research.

   We can now define the *Diverse Subset Problem (DS)* as follows: Given an $m \times n$ array A as defined above and a number $k \leq m$, is there a subset of at least k customers that is diverse?

   Prove that DS is NP-complete.

=====================

Prove versus Show: When an assignment question contains "prove" you are expected to develop a formal proof. There are proof writing resources in *Support Documents*, item "Proofs". When an assignment question contains "show", or no specification at all, you may submit an example, a sketch of a proof, or an argument for your conclusion without meeting the expectations of a proof.