Fall 2023 CSDS 410 Analysis of Algorithms
Prof. Dianne Foreback Assignment 6

**Due Sunday, Dec. 3rd before 11:59 p.m. in Canvas**

**Possible 100 points**

**Directions:**
All work should be your own. You may not copy information from website or use AI tools. This is considered a violation of academic integrity, see https://case.edu/gradstudies/current-students/policies-procedures. If this is violated, you will, at minimum, receive a score of zero on the assignment, it will NOT be dropped as your lowest grade out of the remaining graded assignments, you will be reported per the policy and possible receive a failing grade in the class.

Please upload your Assignment 6 into Canvas and make certain that the quality of the upload is clear. Also, enumerate your answers; e.g. before your answer for problem 1, preceded it with 1a, 1b, 1c, etc.

Please double check that your assignment is properly submitted into Canvas and is visible. Since file dates can be modified, and out of fairness to all, assignments not uploaded into Canvas by the final due date will not be accepted. That is, if your file is on a Google Drive but you forgot to place it into Canvas, it will not be accepted even if the file date shows an acceptable modify date.

Note that this assignment **may** be handwritten providing your writing is easily read.

**Assignment Material:**
This assignment covers complexity theory and NP Completeness.

Submit your work on canvas. Always check canvas for updates and corrections. Unless otherwise stated, whenever a question asks you to **describe an algorithm**, you should:

a. Explain the main concept of your algorithm.
b. Give pseudo-code.
c. Present an example of running your algorithm.
d. Prove/justify its correctness and its running time.

**If you are proving a problem is in NP-Complete, you must**

a. Prove that the problems is in NP (state the verification problem, give and explain an algorithm to verify a certificate in polynomial time, prove its runtime is polynomial time)
b. Prove the problem is in NP-Hard. Do this by giving a reduction of an NP-Complete problem (this problem is given to you) and prove that it reduces to the unknown (the problem you are proving to be in NP-Complete) problem in polynomial time. Then, reduce the unknown to the known given problem in polynomial time. Remember that within the proof of correctness of the reduction to prove in both directions (it is an if and only if). See lectures on how to do this and provide the necessary information.

Please label your problem answers with 1a. 1b. 1c. 1d as appropriate.

Point values for a problem are placed in brackets next to the problem number; e.g. 1 [8] means problem 1 is worth 8 points.

**Problems:**

1. [5] Professor Amongus has shown that a decision problem L is polynomial-time reducible to an *NP-Complete* problem $M$. Moreover, after 80 pages of dense mathematics, he has also just proven that $L$ can be solved in polynomial time. Has he just proven that $P = NP$ ? Why, or why not?

2. [20] Prove that the problem SAT, which takes an arbitrary Boolean formula S as input and asks whether S is satisfiable, is *NP-complete.* Review the instructions at the top of this assignment for "proving a problem is in NP-Complete."

3. [6] Show that the Set-Cover problem is in NP.

4. [6] Given B = $( x_1 \vee \overline{x_2} \vee x_3) \wedge ( x_4 \vee x_5 \vee \overline{x_6} ) \wedge ( x_1 \vee \overline{x_4} \vee \overline{x_5} ) \wedge ( x_3 \vee x_4 \vee x_6 )$
draw the instance of VERTEX-COVER that is constructed by the reduction from 3SAT of the Boolean formula $B$.

5. [6] Draw an example of a graph with 10 vertices and 15 edges that has a vertex cover of size 2.

6. [6] Draw an example of a graph with 10 vertices and 15 edges that has a clique of size 6.

7. [6] Professor Amongus has just designed an algorithm that can take any graph G with n vertices and determine in $O(n^k)$ time whether G contains a clique of size k. Does Professor Amongus deserve the Turing Award for having just shown that ***P = NP***? Why or why not? Hint: look up the definition for a asymptotic polynomial runtime.

8. [15] Consider the 2SAT version of the CNF-SAT problem, in which every clause in the given formula S has exactly two literals. Note that any clause of the form $(a \vee b)$ can be thought of as two implications, $(\overline{a} \rightarrow b)$ and $(\overline{b} \rightarrow a)$. Consider a graph G from S, such that each vertex in G is associated with a variable, x, in S, or its negation, $\overline{x}$ . Let there be a directed edge in G from $a$ $to$ $b$ for each clause equivalent to $(a \rightarrow b)$. Show that S is not satisfiable if and only if there is a variable $x$ such that there is a path in G from $x$ $to$ $\overline{x}$ and a path from $\overline{x}$ $to$ $x$. Derive from this rule a polynomial-time algorithm for solving this special case of the CNF-SAT problem. What is the running time of your algorithm?

9. [10] Suppose an oracle has given you a magic computer, *C*, that when given any Boolean formula *B* in CNF will tell you in one step whether *B* is satisfiable. Show how to use *C* to construct an actual assignment of satisfying Boolean values to the variables in any satisfiable formula *B*. How many calls do you need to make to *C* in the worst case in order to do this? **Hint**: Consider dealing with each variable one at a time.

10. [20] Define SUBGRAPH-ISOMORPHISM as the problem that takes a graph, $G$, and another graph, $H$, and determines if H is isomorphic to a subgraph of $G$. That is, the problem is to determine whether there is a one-to-one mapping, f, of the vertices in $H$ to a subset of the vertices in $G$ such that, if $(v, w)$ is an edge in H, then $(f(v), f(w))$ is an edge in $G$. Prove that SUBGRAPH-ISOMORPHISM is NP-complete. In your proof **you must show** a reduction of the Hamiltonian Cycle to this problem. Please make certain you follow instructions at the top of this assignment on proving a problem is NP-Complete.