Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

Problem 1. (10 points) Show that the language
$L = \{< M, w, k > : TM\ M \text{ accepts input } w \text{ and never moves its head beyond the first } k \text{ tape cells}\}$
is decidable.

L = {<M, w, k> : *TM M* accepts input w and never moves its head beyond the first k tape cells}

TM M to decide L:

On input <M, w, k>

1. Let M_1 be a turing machine that simulates M.
2. Set M_1 to have k number of tape cells.
3. If M_1 on input w moves it head reading w and stops before or at most at the kth tape cell, then M accepts.
4. If M_1 on input w attempts to move it head past the kth tape cell, then M rejects.
5. Since M rejects when M_1 attempts to move its head after the kth tape cell and has an accept state, it is decidable.

Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

Problem 2. (10 points) Recall that $A <_p B$ if there is a polynomial-time computable function f such that $w \in A \Leftrightarrow f(w) \in B$.

a. Show that the relation $<_p$ over languages is transitive.
b. Show that if $\forall A, B \in \boldsymbol{P}$, if $B \neq \phi$ and $B \neq \Sigma^*$ then $A <_p B$.
(Hint: this is easier than it seems since $A \in \boldsymbol{P}$. Now use the definition.)
c. Show that if $\boldsymbol{P} = \boldsymbol{NP}$ then every language, other than ϕ and Σ^*, in $\boldsymbol{P}$ is **$\boldsymbol{NP}$-Complete.** (You can use the result of part (b) even if you didn't prove that.)

a. There are polynomial time functions f and g such that $w \in A \Leftrightarrow f(w) \in B$ where $A <_p B$ and $w \in B \Leftrightarrow q(w) \in C$ where $B <_p C$. The function f(w) maps A to B, and q(w) maps B to C. Thus, the function $h = q(f(w))$ maps A to C, or $w \in A \Leftrightarrow q(f(w)) \in C$ where $A <_p C$, meaning the relation is transitive.
b. Because B is not φ or Σ^* we know that the language is somewhere between an empty string language and the sum of all string languages. A is a language decidable in polynomial time and therefore, there are languages in B which can be reduced down to A in polynomial time.
c. If B is any language in P and P=NP, that means B is also in NP and since any problem in NP can be transformed into another problem in NP-complete, every language except φ or Σ^* is NP-complete.

Names: Omar Abdelmotaleb and Benjamin Singleton

Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb

Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

Problem 3. (20 points) Behold, a genie appears before you! Given a formula $\phi(x_1, x_2, \ldots, x_n)$ in conjunctive normal form with n boolean variables, the genie will correctly tell you (in one step) whether or not the formula is satisfiable. Unfortunately, the genie will not give you a truth assignment to the variables that makes the formula true.

Your problem is to figure out a satisfying truth assignment when the genie says the formula is satisfiable. You can present the genie with a polynomial (in n) number of queries.

i. (5 points) Give a high-level description of your algorithm, with enough detail.
ii. (2 points) What is the maximum number of queries made by your algorithm?
iii. (3 points) Explain why your algorithm correctly finds a satisfying assignment for a satisfiable formula.
iv. (10 points) A second genie appears! Given an undirected graph, this genie will correctly tell you whether or not the graph has a Hamiltonian cycle. How will you use this genie to find a Hamiltonian cycle in any graph that has one in polynomial time?

i. Implying that the genie said the initial formula is satisfiable, set x_1 to be true and ask him again. If he says yes, then x_1 is true, if he says no, then x_1 is false. Move to x_2 and repeat the process. Do this process for all variables in the formula, after which, you will know the truth assignment for all the variables.

ii. At most, you'd have to ask the genie about the status of every single variable, in which case the maximum number of queries would equal to the number of variables which is n.

iii. My algorithm correctly finds a satisfying assignment for a satisfiable formula because it determines the definitive status of each given variable. Because they are boolean variables, each variable only has two states and similarly the formula only has two types of assignments, satisfiable or unsatisfiable. When we set x_1 to true and ask the genie about the formula, his answer tells us indirectly the definitive assignment of that variable. If he says the formula is satisfiable, then we know that x_1 must be set to true for the formula to be satisfiable and inversely, if he says no then we know that x_1 must be false since those are the only two options for a boolean variable. By going through all of the variables with this logic I know that my algorithm is correctly finding a satisfying assignment for a satisfiable formula.

iv. To find a hamiltonian cycle in a graph which ones in polynomial time, I will start by choosing a random edge in the graph and removing it. I will ask the genie if there is still a hamiltonian cycle in the graph without the edge. If there is no longer a hamiltonian cycle in the graph, that means the edge is necessary for the cycle and so I add it back in. If there isn't, that means the edge isn't part of the cycle and so I permanently remove it. I repeat this process for all edges in the graph. When I am done applying this algorithm, the edges of the graph which remain will be the hamiltonian cycle.

Names: Omar Abdelmotaleb and Benjamin Singleton

Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb

Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

Problem 4. (10 points)

(a) Give a high-level description of a linear time algorithm to determine if a directed graph contains a directed cycle.

(b) Next, suppose you are given a formula in 2CNF with n variables $x_1, \ldots, x_n$. Construct a graph with $2n$ vertices, one for each literal, and for every clause construct two edges as follows:
If the clause is of the form $(x_i \vee x_j)$, $add\ the\ directed\ edges\ (\neg x_i, x_j)\ and\ (\neg x_j, x_i)$.
If the clause is of the form $(\neg x_i \vee x_j), add\ directed\ edges\ (x_i, x_j)\ and\ (\neg x_j, \neg x_i)$.
In general, for the clause is $(a, b), add\ directed\ edges\ (\neg a, b)\ and\ (\neg b, a)$.
Describe how you would use your algorithm from part (a) to determine if the 2CNF formula is satisfiable and prove that your algorithm for satisfiability is correct.

(a) You can determine if a directed graph contains a directed cycle by utilizing a breadth-first-search algorithm which runs in linear time. You first determine the in-degree for each vertex. This can be accomplished by traversing over an array of edges and increasing the counter for the destination vertex by 1. This is a linear process that takes θ(V+E) time where V is the number of vertices and E is the number of edges, making this process linear. Then, create a counter for the number of vertices visited and add the all vertices with an in-degree of 0 to a queue. Similarly, this process also takes linear time. Next, remove the first vertex from the queue and then increment the count of visited nodes by 1, decrease the in-degree of all its neighbors by 1, and if a neighbors in-degree becomes 0 add it to the queue. Repeat this process until the queue is empty. If the counter does not equal the number of nodes then the graph has a cycle, otherwise it does not.

(b) We use the breadth-first-search from part (a) to determine if there is a directed cycle in the graph. If there is no directed cycle in the graph, then the 2CNF formula is satisfiable. This is because if there was a directed cycle, then an additional clause would be formed as a result and would invalidate the formula.