Due: July 31, in tutorial[1]

1. The *resolvent* of the clauses $C_1 \vee x \vee C_2$ and $D_1 \vee \bar{x} \vee D_2$ is the clause $C_1 \vee C_2 \vee D_1 \vee D_2$, with repeated variables simplified. For example, if we have the following two clauses: $x \vee \bar{y} \vee z \vee w$ and $x \vee v \vee z \vee y$, then the resolvent is $x \vee z \vee w \vee x \vee v \vee z$, which simplifies to $x \vee z \vee w \vee v$.

   Fact: A set S of clauses is unsatisfiable iff the empty clause $\emptyset$ can be derived from S by finitely many resolvents. For example, if $S = \{p \vee q, \bar{q} \vee r, \bar{p} \vee r, \bar{p} \vee \bar{r}, p \vee \bar{r}\}$, then $\emptyset$ can be derived as follows:

1	$p \vee q$	
2	$\bar{q} \vee r$	
3	$\bar{p} \vee r$	
4	$\bar{p} \vee \bar{r}$	
5	$p \vee \bar{r}$	
6	$p \vee r$	from 1 and 2
7	r	from 3 and 6
8	$\bar{r}$	from 4 and 5
9	$\emptyset$	from 7 and 8

   (a) Use the **Fact** to show that 2-SAT is in P. That is, give a polytime algorithm which, given a set S of 2-literal clauses, determines whether S is satisfiable. Estimate the running time of your algorithm in terms of m, the number of clauses in S.

   (b) What goes wrong when you try to adapt your algorithm from (a) to give a polytime algorithm for 3-SAT?

2. For $k \in \mathbb{N}$ a fixed constant, we define the problem k-COL as follows:
   Input: Graph $G = (V, E)$.
   Question: Can the nodes of G be colored using at most k colors in such a way that adjacent nodes have different colors?

   Note that k is not part of the input; 2-COL, 3-COL, 4-COL, etc., are all different problems.

   (a) Show that 2-COL is in P.

   (b) Use the fact that 3-COL is NP-complete to show that k-COL is NP-complete, for all $k \geq 3$.

[1] Assignments are due at the *beginning* of tutorial, and since solutions are discussed in tutorial, late assignments are *not* accepted. The work that you submit *must* be your own. You may discuss problems with each other; however, you should take no notes during these discussions, and you should prepare written solutions alone.

3. For each of the following problems, state whether or not it has a polytime algorithm, and justify your answer (give a brief outline of the algorithm or the reason why there is none—assume P $\neq$ NP, and you can refer to the NP-complete problems given in the notes).

   (a) EVEN-SAT:
   Input: A set S of clauses, where each clause has an even number of literals.
   Question: Is S satisfiable?

   (b) UNARY-TRUCK-ROUTING:
   Input: An undirected graph $G = (V, E)$, a node $v_0 \in V$, a positive length $l(e)$ (presented in unary notation) assigned to each edge $e \in E$, and positive integers D and K.
   Question: Is there an assignment of routes to each of K trucks such that each node in V is visited by at least one truck and the total combined distance traveled by all trucks is at most D? Each route must start and end at node v_0, and visit some sequence of nodes (possibly with repetition) by following edges in E.

   (c) BINARY-SINGLE-TRUCK-DESTINATION:
   Input: As undirected graph $G = (V, E)$, nodes s, t in V, a positive integer length $l(e)$ (presented in binary notation) assigned to each edge $e \in E$, and a positive integer B.
   Question: Is there a route in G from s to t of total length at most B? The route must follow edges in E.

   (d) BINARY-LARGE-TREE:
   Input: An undirected graph $G = (V, E)$, a positive integer value $c(e)$ (presented in binary notation) assigned to each edge $e \in E$, and a positive integer B.
   Question: Does G have a spanning tree of total value B or more?

   (e) SEATING-DIPLOMATS:
   Input: A sequence of k diplomats; for each (unordered) pair of diplomats, we are given whether or not they like each other.
   Question: Is it possible to seat the diplomats around a circular table, in such a way that every two diplomats who are seated next to each other, like each other?

4. Let $f : \Sigma^* \longrightarrow \Sigma^*$ be a computable function satisfying the condition that for all x and y in Σ^*:

$$|x| \leq |y| \Longrightarrow |f(x)| \leq |f(y)|$$

   Show that the range of f (i.e., the set $\{f(x) | x \in \Sigma^*\}$) is a decidable set.

5. Consider the language $L = \{\langle M_1, M_2 \rangle |$ where M_1 and M_2 are Turing Machines for which there exists an input w such that M_1 accepts w and M_2 rejects $w\}$.

   (a) Show that L is semi-decidable.

   (b) Show that L is undecidable.