Describe each problem with a linear or integer program.

1. Simple Scheduling with Prerequisites (SSP):
Given a set of problems, some of which need to be finished in order to begin others, give start times to the jobs which allow all of the prerequites to be met (if there are circular prerequisites, then no solution is possible). More precisely, we are given a list of durations $d_1, d_2, ..., d_n$ for each job, together with prerequisites between the jobs, i.e., values $p_{i,j}$ for $1 \leqslant i, j \leqslant n$ such that $p_{i,j} = 1$ if job i is a prerequisite of job j; $p_{i,j} = 0$ otherwise. We want to find start times $s_1, s_2, ..., s_n$ for each job such that each job i finished no later than the start time of all jobs that have i as a prerequisite.

Solution: For each job, we have 1 variable s_i = start time of job i. In addition, we have one more variable T = time to completion.

Minimize time to completion T

Subject to:

$T \geqslant s_i + d_i$ for $i = 1, 2, ..., n$

$s_i \geqslant p_{j,i} \times (s_j + d_j)$ for $i = 1, 2, ..., n; j = 1, 2, ..., n$

The constraints ensure that each job starts only after each of its prerequisites has completed, and that the total completion time has the correct value. The fact that we try to minimize T ensures that none of the values in a solution are greater than they need to be.

---

2. A problem on Sets:
Given a set of elements $E = \{x_1, x_2, ..., x_n\}$ and a set of subsets of E, $S = \{H_1, H_2, ..., H_m\}$, we want to find the smallest subset C of E such that for each set H_i, $C \cap H_i \neq \varnothing$.

Solution: We have one variable v_i for each element of E.

Minimize $\sum_{i=1}^{n} v_i$

Subject to:

$v_i \in \{0, 1\}$ for each element x_i

$v_a + v_b + ... + v_k \geqslant 1$ for each $H_j = \{x_a, x_b, ..., x_k\}$

Let $C = \{x_i : v_i = 1\}$.

Minimization ensures that we pick as few values $v_i = 1$ as possible. Constraints ensure that at least one element is picked for each set.
NOTE: This is Integer Programming (restricting solution to integer values), which is not strictly the same as Linear Programming.

---

3. Satisfiablity (SAT):
Given a formula F in CNF, we want to know if F is satisfiable, i.e., if there is some setting of the variables of F that makes F true.

F has the form $C_1 \wedge C_2 \wedge ... \wedge C_r$, where each clause C_j is a disjunction of one or more literals, $C_j = (a_{j,1} \vee ... \vee a_{j,s_j})$, with each a_i in $\{v_1, \sim v_1, ..., v_n, \sim v_n\}$. For example,

$$F = (v_1 \vee v_2) \wedge (\sim v_1 \vee v_3 \vee v_4) \wedge (\sim v_2)$$

Solution: Use one variable x_i for each propositional variable v_i.

Constraints:

$x_i \in \{0, 1\}$ for each x_i

$b_1 + ... + b_s \geqslant 1$ for each clause $(a_1 \vee ... \vee a_s)$,

where $b_j = x_i$ if $a_j = v_i$ and $b_j = (1 - x_i)$ if $a_j = \sim v_i$.

For example, for the formula above,

$$x_1, x_2, x_3, x_4 \in \{0, 1\}$$

$$x_1 + x_2 \geqslant 1$$

$$(1 - x_1) + x_3 + x_4 \geqslant 1$$

$$1 - x_2 \geqslant 1$$

Objective function? None! We only care whether or not the constraints can be satisfied as they correspond exactly to the structure of the formula (and whether or not it can be satisfied).

To get answer to original question, set v_i = true iff $x_i = 1$. Any assignment of values to the variables of F making F true must make at least one literal true in each clause of F, and this corresponds to a setting of the variables of the integer program that satisfies each constraint. The converse is also true because of the way the integer program constraints are written based on the formula.