1. **Aping Simplex**
Solve the following linear programming using the simplex algorithm. You may assume that you have already completed phase 1 and know the vertex \( (1,0) \) to start with.

\[
\text{max } 3x + y \quad \text{subject to,}
\]

\[
x + y \geq 1 \\
y - x \geq 1 \\
x \leq 3 \\
x, y \geq 0
\]

2. **Dual of max-flow**

   a) Write down the problem of finding the maximum flow from \( s \) to \( t \) in a directed network as a linear program.

   b) Write the dual of this linear program using a variable \( y_e \) for each edge and \( x_u \) for each vertex \( u \neq s, t \).

   c) Show that for any solution to the general dual LP, the sum of the \( y_e \) values along any directed path from \( s \) to \( t \) must be at least 1.

   d) Interpret the dual variables \( y_e \) as length of a pipe placed along the edge \( e \), with area \( c_e \). Give a consistent interpretation for the objective function. Show using this interpretation that any solution to this LP gives an upper bound on the maximum flow.

   e) Show that any \( s-t \) cut in the graph can be translated into a dual feasible solution whose cost is exactly the capacity of the cut.

3. **Dual of shortest paths**
Suppose we want to compute the shortest path from \( s \) to \( t \) in a directed graph with edge lengths \( l_e > 0 \).

   a) Show that this is equivalent to finding an \( s-t \) flow \( f \) that minimizes \( \sum_e l_e f_e \) subject to \( \text{size}(f) = 1 \). There are no capacity constraints.

   b) Write the shortest path problem as a linear program.

   c) Show that the dual LP can be written as

   \[
   \max x_s - x_t \\
x_u - x_v \leq l_{uv} \text{ for all } (u, v) \in E
   \]

   d) Interpret this dual for undirected graphs.
4. Reduce, Reduce, Reduce...
Give simple reductions for the following:

- 3SAT to Integer Linear Programming
- Independent Set to Clique
- Independent Set to Vertex Cover
- Vertex Cover to Set Cover

5. Finding Nemo (or SAT solutions if you want)
Given an algorithm that decides whether a given SAT instance, there exists a truth assignment satisfying it, give an algorithm for finding a satisfying assignment.