29 Linear Programming

29.1 Standard and slack forms

29.1-1

> If we express the linear program in (29.24)–(29.28) in the compact notation of (29.19)–(29.21), what are n, m, A, b, and c?

$$n = m = 3,$$
$$A = \begin{pmatrix} 1 & 1 & -1 \\ -1 & -1 & 1 \\ 1 & -2 & 2 \end{pmatrix},$$
$$b = \begin{pmatrix} 7 \\ -7 \\ 4 \end{pmatrix},$$
$$c = \begin{pmatrix} 2 \\ -3 \\ 3 \end{pmatrix}.$$

29.1-2

> Give three feasible solutions to the linear program in (29.24)–(29.28). What is the objective value of each one?

1. $(x_1, x_2, x_3) = (6, 1, 0)$ with objective value 9.
2. $(x_1, x_2, x_3) = (5, 2, 0)$ with objective value 4.
3. $(x_1, x_2, x_3) = (4, 3, 0)$ with objective value -1.

29.1-3

> For the slack form in (29.38)–(29.41), what are N, B, A, b, c, and v?

$$N = \{1, 2, 3\},$$
$$B = \{4, 5, 6\},$$
$$A = \begin{pmatrix} 1 & 1 & -1 \\ -1 & -1 & 1 \\ 1 & -2 & 2 \end{pmatrix},$$
$$b = \begin{pmatrix} 7 \\ -7 \\ 4 \end{pmatrix},$$
$$c = \begin{pmatrix} 2 \\ -3 \\ 3 \end{pmatrix},$$
$$v = 0.$$

29.1-4

> Convert the following linear program into standard form:

$$\begin{array}{llllll}
\text{minimize} & 2x_1 & + & 7x_2 & + & x_3 \\
\text{subject to} & & & & & \\
& x_1 & & & - & x_3 & = & 7 \\
& 3x_1 & + & x_2 & & & \geq & 24 \\
& & & x_2 & & & \geq & 0 \\
& & & & & x_3 & \leq & 0 \;.
\end{array}$$

$$\begin{array}{llllllll}
\text{maximize} & -2x_1 & & - & 2x_2 & - & 7x_3 & + & x_4 \\
\text{subject to} & & & & & & & \\
& -x_1 & + & x_2 & & & - & x_4 & \leq & -7 \\
& x_1 & - & x_2 & & & + & x_4 & \leq & 7 \\
& -3x_1 & + & 3x_2 & - & x_3 & & & \leq & -24 \\
& x_1, x_2, x_3, x_4 & & & & & & & \leq & 0 \;.
\end{array}$$

29.1-5

Convert the following linear program into slack form:

$$\begin{array}{llllllll}
\text{maximize} & 2x_1 & & & - & 6x_3 \\
\text{subject to} & & & & & \\
& x_1 & + & x_2 & - & x_3 & \leq & 7 \\
& 3x_1 & - & x_2 & & & \geq & 8 \\
& -x_1 & + & 2x_2 & + & 2x_3 & \geq & 0 \\
& x_1, x_2, x_3 & & & & & \geq & 0 \;.
\end{array}$$

What are the basic and nonbasic variables?

First, we will multiply the second and third inequalities by minus one to make it so that they are all $\leq$ inequalities. We will introduce the three new variables x_4, x_5, x_6, and perform the usual procedure for rewriting in slack form

$$\begin{array}{lllllllll}
x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\
x_5 & = & -8 & + & 3x_1 & - & x_2 & & \\
x_6 & = & & - & x_1 & + & 2x_2 & + & 2x_3 \\
x_1, x_2, x_3, x_4, x_5, x_6 & \geq & 0 & , & & & &
\end{array}$$

where we are still trying to maximize $2x_1 - 6x_3$. The basic variables are x_4, x_5, x_6 and the nonbasic variables are x_1, x_2, x_3.

29.1-6

Show that the following linear program is infeasible:

$$\begin{array}{llllll}
\text{minimize} & 3x_1 & - & 2x_2 \\
\text{subject to} & & & \\
& x_1 & + & x_2 & \leq & 2 \\
& -2x_1 & - & 2x_2 & \leq & -10 \\
& x_1, x_2 & & & \geq & 0 \;.
\end{array}$$

By dividing the second constraint by 2 and adding to the first, we have $0 \leq -3$, which is impossible. Therefore there linear program is unfeasible.

29.1-7

Show that the following linear program is unbounded:

$$\begin{array}{rlrlcr}
\text{minimize} & x_1 & - & x_2 \\
\text{subject to} \\
& -2x_1 & + & x_2 & \leq & -1 \\
& -x_1 & - & 2x_2 & \leq & -2 \\
& x_1, x_2 & & & \geq & 0 \quad.
\end{array}$$

For any number $r > 1$, we can set $x_1 = 2r$ and $x_2 = r$. Then, the restaints become

$$\begin{array}{rcrcl}
-2(2r) & + & r = -3r & \leq & -1 \\
-2r & - & 2r = -4r & \leq & -2 \\
2r, r & & & \geq & 0 \quad.
\end{array}$$

All of these inequalities are clearly satisfied because of our initial restriction in selecting r. Now, we look to the objective function, it is $2r - r = r$. So, since we can select r to be arbitrarily large, and still satisfy all of the constraints, we can achieve an arbitrarily large value of the objective function.

29.1-8

> Suppose that we have a general linear program with n variables and m constraints, and suppose that we convert it into standard form. Give an upper bound on the number of variables and constraints in the resulting linear program.

In the worst case, we have to introduce 2 variables for every variable to ensure that we have nonnegativity constraints, so the resulting program will have $2n$ variables. If each constraint is an equality, we would have to double the number of constraints to create inequalities, resulting in $2m$ constraints. Changing minimization to maximization and greater-than signs to less-than signs don't affect the number of variables or constraints, so the upper bound is $2n$ on variables and $2m$ on constraints.

29.1-9

> Give an example of a linear program for which the feasible region is not bounded, but the optimal objective value is finite.

Consider the linear program where we want to maximize $x_1 - x_2$ subject to the constraints $x_1 - x_2 \leq 1$ and $x_1, x_2 \geq 0$. clearly the objective value can never be greater than one, and it is easy to achieve the optimal value of 1, by setting $x_1 = 1$ and $x_2 = 0$. Then, this feasible region is unbounded because for any number r, we could set $x_1 = x_2 = r$, and that would be feasible because the difference of the two would be zero which is ≤ 1.

If we further wanted it so that there was a single solution that achieved the finite optimal value, we could add the requirements that $x_1 \leq 1$.

29.2 Formulating problems as linear programs

29.2-1

> Put the single-pair shortest-path linear program from (29.44)–(29.46) into standard form.

The objective is already in normal form. However, some of the constraints are equality constraints instead of $\leq$ constraints. This means that we need to rewrite them as a pair of inequality constraints, the overlap of whose solutions is just the case where we have equality. we also need to deal with the fact that most of the variables can be negative. To do that, we will introduce variables for the negative part and positive part, each of which need be positive, and we'll just be sure to subtract the negative part. d_s need not be changed in this way since it can never be negative since we are not assuming the existence of negative weight cycles.

$$d_v^+ - d_v^- - d_u^+ + d_u^- \leq w(u, v) \text{ for every edge } (u, v)$$
$$d_s \leq 0$$

29.2-2

> Write out explicitly the linear program corresponding to finding the shortest path from node s to node y in Figure 24.2(a).

$$\text{minimize} \quad d_y$$
$$\text{subject to}$$
$$d_t \leq d_s + 3$$
$$d_x \leq d_t + 6$$
$$d_y \leq d_s + 5$$
$$d_y \leq d_t + 2$$
$$d_z \leq d_x + 2$$
$$d_t \leq d_y + 1$$
$$d_x \leq d_y + 4$$
$$d_z \leq d_y + 1$$
$$d_s \leq d_z + 1$$
$$d_x \leq d_z + 7$$
$$d_2 = 0.$$

29.2-3

> In the single-source shortest-paths problem, we want to find the shortest-path weights from a source vertex s to all vertices $v \in V$. Given a graph G, write a linear program for which the solution has the property that d_v is the shortest-path weight from s to v for each vertex $v \in V$.

We will follow a similar idea to the way to when we were finding the shortest path between two particular vertices.

$$\text{maximize} \quad \sum_{v \in V} d_v$$
$$\text{subject to}$$
$$d_v \leq d_u + w(u, v) \text{ for each edge } (u, v)$$
$$d_s = 0.$$

The first type of constraint makes sure that we never say that a vertex is further away than it would be if we just took the edge corresponding to that constraint. Also, since we are trying to maximize all of the variables, we will make it so that there is no slack anywhere, and so all the dv values will correspond to lengths of shortest paths to v. This is because the only thing holding back the variables is the information about relaxing along the edges, which is what determines shortest paths.

29.2-4

> Write out explicitly the linear program corresponding to finding the maximum flow in Figure 26.1(a).

$$
\begin{aligned}
\text{maximize} \quad & f_{sv_1} + f_{sv_2} \\
\text{subject to} \quad & \\
& f_{sv_1} \le 16 \\
& f_{sv_2} \le 14 \\
& f_{v_1 v_3} \le 12 \\
& f_{v_2 v_1} \le 4 \\
& f_{v_2 v_4} \le 14 \\
& f_{v_3 v_2} \le 9 \\
& f_{v_3 t} \le 20 \\
& f_{v_4 v_3} \le 7 \\
& f_{v_4 t} \le 4 \\
& f_{sv_1} + f_{v_2 v_1} = f_{v_1 v_3} \\
& f_{sv_2} + f_{v_3 v_2} = f_{v_2 v_1} + f_{v_2 v_4} \\
& f_{v_1 v_3} + f_{v_4 v_3} = f_{v_3 v_2} + f_{v_3 t} \\
& f_{v_2 v_4} = f_{v_4 v_3} + f_{v_4 t} \\
& f_{uv} \ge 0 \text{ for } u, v \in \{s, v_1, v_2, v_3, v_4, t\}.
\end{aligned}
$$

29.2-5

> Rewrite the linear program for maximum flow (29.47)–(29.50) so that it uses only $O(V + E)$ constraints.

All we need to do to bring the number of constraints down from $O(V^2)$ to $O(V + E)$ is to replace the way we index the flows. Instead of indexing it by a pair of vertices, we will index it by an edge. This won't change anything about the analysis because between pairs of vertices that don't have an edge between them, there definitely won't be any flow. Also, it brings the number of constraints of the first and third time down to $O(E)$ and the number of constraints of the second kind stays at $O(V)$.

$$
\begin{aligned}
\text{maximize} \quad & \sum_{\text{edges } e \text{ leaving } s} f_e - \sum_{\text{edges } e \text{ entering } s} f_s \\
\text{subject to} \quad & \\
& f_{(u,v)} \le c(u, v) \text{ for each edge } (u, v) \\
& \sum_{\text{edges } e \text{ leaving } u} f_e - \sum_{\text{edges } e \text{ entering } u} f_e \text{ for each edge } u \in V - \{s, t\} \\
& f_e \ge 0 \text{ for each edge } e.
\end{aligned}
$$

29.2-6

> Write a linear program that, given a bipartite graph $G = (V, E)$ solves the maximum-bipartite-matching problem.

Recall from section 26.3 that we can solve the maximum-bipartite-matching problem by viewing it as a network flow problem, where we append a source s and sink t, each connected to every vertex is L and R respectively by an edge with capacity 1, and we give every edge already in the bipartite graph capacity 1. The integral maximum flows are in correspondence with maximum bipartite matchings. In this setup, the linear programming problem to solve is as follows:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{v \in L} f_{sv} \\
\text{subject to} \quad & \\
& f_{(u,v)} \le 1 \text{ for each } u, v \in \{s\} \cup L \cup R \cup \{t\} = V \\
& \sum_{v \in V} f_{vu} = \sum_{v \in V} f_{uv} \text{ for each } u \in L \cup R \\
& f_{uv} \ge 0 \text{ for each } u, v \in V
\end{aligned}
$$

29.2-7

> In the ***minimum-cost multicommodity-flow problem***, we are given directed graph $G = (V, E)$ in which each edge $(u, v) \in E$ has a nonnegative capacity $c(u, v) \geq 0$ and a cost $a(u, v)$. As in the multicommodity-flow problem, we are given k different commodities, $K_1, K_2, \ldots, K_k$, where we specify commodity i by the triple $K_i = (s_i, t_i, d_i)$. We define the flow f_i for commodity i and the aggregate flow f_{uv} on edge (u, v) as in the multicommodity-flow problem. A feasible flow is one in which the aggregate flow on each edge (u, v) is no more than the capacity of edge (u, v). The cost of a flow is $\sum_{u,v \in V} a(u, v) f_{uv}$, and the goal is to find the feasible flow of minimum cost. Express this problem as a linear program.

As in the minimum cost flow problem, we have constraints for the edge capacities, for the conservation of flow, and nonegativity. The difference is that the restraint that before we required exactly d units to flow, now, we require that for each commodity, the right amount of that commodity flows. the conservation equalities will be applied to each different type of commodity independently. If we super script f that will denote the type of commodity the flow is describing, if we do not superscript it, it will denote the aggregate flow

We want to minimize

$$\sum_{u,v \in V} a(u, v) f_{uv}.$$

The capacity constraints are that

$$\sum_{i \in [k]} \sum_{u,v \in V} f^i_{uv} \leq c(u, v).$$

The conservation constraints are that for every $i \in [k]$, for every $u \in V \setminus \{s_i, t_i\}$.

$$\sum_{v \in V} f^i_{uv} = \sum_{v \in V} f^i_{vu}.$$

Now, the constraints that correspond to requiring a certain amount of flow are that for every $i \in [k]$.

$$\sum_{v \in V} f^i_{s_i,v} - \sum_{v \in V} f^i_{v,s_i} = d.$$

Now, we put in the constraint that makes sure what we called the aggregate flow is actually the aggregate flow, so, for every $u, v \in V$,

$$f_{uv} = \sum_{i \in [k]} f^i_{uv}.$$

Finally, we get to the fact that all flows are nonnegative, for every $u, v \in V$,

$$f_{uv} \geq 0.$$

29.3 The simplex algorithm

29.3-1

> Complete the proof of Lemma 29.4 by showing that it must be the case that $c = c'$ and $v = v'$.

We subtract equation (29.81) from equation (29.79).

$$z = v + \sum_{j \in N} c_j x_j, \qquad (29.79)$$

$$z = v' + \sum_{j \in N} c'_j x_j. \qquad (29.81)$$

Thus we have,

$$0 = v - v' + \sum_{j \in N} (c_j - c_j')x_j.$$

$$\sum_{j \in N} c_j' x_j = v - v' + \sum_{j \in N} c_j x_j.$$

By Lemma 29.3, we have $c_j = c_j'$ for every j and $v = v'$ since $v - v' = 0$.

29.3-2

> Show that the call to PIVOT in line 12 of SIMPLEX never decreases the value of v.

The only time v is updated in PIVOT is line 14, so it will suffice to show that $c_e \hat{b}_e \geq 0$. Prior to making the call to PIVOT, we choose an index e such that $c_e > 0$, and this is unchanged in PIVOT. We set $\hat{b}_e$ in line 3 to be b_l/a_{le}.

The loop invariant proved in Lemma 29.2 tells us that $b_l \geq 0$. The if-condition of line 6 of SIMPLEX tells us that only the noninfinite δ_i must have $a_{ie} > 0$, and we choose l to minimize δ_l, so we must have $a_{le} > 0$. Thus, $c_e \hat{b}_e \geq 0$, which implies v can never decrease.

29.3-3

> Prove that the slack form given to the PIVOT procedure and the slack form that the procedure returns are equivalent.

To show that the two slack forms are equivalent, we will show both that they have equal objective functions, and their sets of feasible solutions are equal.

First, we'll check that their sets of feasible solutions are equal. Basically all we do to the constraints when we pivot is take the non-basic variable, e, and solve the equation corresponding to the basic variable l for e. We are then taking that expression and replacing e in all the constraints with this expression we got by solving the equation corresponding to l. Since each of these algebraic operations are valid, the result of the sequence of them is also algebraically equivalent to the original.

Next, we'll see that the objective functions are equal. We decrease each c_j by $c_e \hat{a}_{ej}$, which is to say that we replace the non-basic variable we are making basic with the expression we got it was equal to once we made it basic.

Since the slack form returned by PIVOT, has the same feasible region and an equal objective function, it is equivalent to the original slack form passed in.

29.3-4

> Suppose we convert a linear program (A, b, c) in standard form to slack form. Show that the basic solution is feasible if and only if $b_i \geq 0$ for $i = 1, 2, \ldots, m$.

First suppose that the basic solution is feasible. We set each $x_i = 0$ for $1 \leq i \leq n$, so we have $x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij}x_j = b_i$ as a satisfied constraint. Since we also require $x_{n+i} \geq 0$ for all $1 \leq i \leq m$, this implies $b_i \geq 0$.

Now suppose $b_i \geq 0$ for all i. In the basic solution we set $x_i = 0$ for $1 \leq i \leq n$ which satisfies the nonnegativity constraints. We set $x_{n+i} = b_i$ for $1 \leq i \leq m$ which satisfies the other constraint equations, and also the nonnegativity constraints on the basic variables since $b_i \geq 0$. Thus, every constraint is satisfied, so the basic solution is feasible.

29.3-5

> Solve the following linear program using SIMPLEX:

$$
\begin{aligned}
\text{maximize} \quad & 18x_1 \;+\; 12.5x_2 \\
\text{subject to} \quad & \\
& x_1 \;+\; x_2 \;\le\; 20 \\
& x_1 \le\; 12 \\
& x_2 \;\le\; 16 \\
& x_1, x_2 \ge\; 0 \quad .
\end{aligned}
$$

First, we rewrite the linear program into it's slack form

$$
\begin{aligned}
\text{maximize} \quad & 18x_1 + 12.5x_2 \\
\text{subject to} \quad & \\
x_3 \;&=\; 20 - x_1 - x_2 \\
x_4 \;&=\; 12 - x_1 \\
x_5 \;&=\; 16 - x_2 \\
x_1, x_2, x_3, x_4, x_5 \;&\ge 0.
\end{aligned}
$$

We now stop since no more non-basic variables appear in the objective with a positive coefficient. Our solution is $(12, 8, 0, 0, 8)$ and has a value of 316. Going back to the standard form we started with, we just disregard the values of x_3 through x_5 and have the solution that $x_1 = 12$ and $x_2 = 8$. We can check that this is both feasible and has the objective achieve 316.

29.3-6

Solve the following linear program using SIMPLEX:

$$
\begin{aligned}
\text{maximize} \quad & 5x_1 \;-\; 3x_2 \\
\text{subject to} \quad & \\
& x_1 \;-\; x_2 \;\le\; 1 \\
& 2x_1 \;+\; x_2 \;\le\; 2 \\
& x_1, x_2 \ge\; 0 \quad .
\end{aligned}
$$

First, we convert the linear program into it's slack form

$$
\begin{aligned}
z \;&=\; 5x_1 \;-\; 3x_2 \\
x_3 \;&=\; 1 \;-\; x_1 \;+\; x_2 \\
x_4 \;&=\; 2 \;-\; 2x_1 \;-\; x_2 \quad .
\end{aligned}
$$

The nonbasic variables are x_1 and x_2. Of these, only x_1 has a positive coefficient in the objective function, so we must choose $x_e = x_1$. Both equations limit x_1 by 1, so we'll choose the first one to rewrite x_1 with. Using $x_1 = 1 - x_3 + x_2$ we obtain the new system

$$
\begin{aligned}
z \;&=\; 5 \;-\; 5x_3 \;+\; 2x_2 \\
x_1 \;&=\; 1 \;-\; x_3 \;+\; x_2 \\
x_4 \;&=\; 2x_3 \;-\; 3x_2 \quad .
\end{aligned}
$$

Now x_2 is the only nonbasic variable with positive coefficient in the objective function, so we set $x_e = x_2$. The last equation limits x_2 by 0 which is most restrictive, so we set $x_2 = \frac{2}{3}x_3 - \frac{1}{3}x_4$. Rewriting, our new system becomes

$$
\begin{aligned}
z \;&=\; 5 \;-\; \frac{11}{3}x_3 \;-\; \frac{2}{3}x_4 \\
x_1 \;&=\; 1 \;-\; \frac{1}{3}x_3 \;-\; \frac{1}{3}x_4 \\
x_2 \;&=\; \frac{2}{3}x_3 \;-\; \frac{1}{3}x_4 \quad .
\end{aligned}
$$

Every nonbasic variable now has negative coefficient in the objective function, so we take the basic solution $(x_1, x_2, x_3, x_4) = (1, 0, 0, 0)$. The objective value this achieves is 5.

29.3-7

Solve the following linear program using SIMPLEX:

$$
\begin{array}{llllll}
\text{minimize} & x_1 & + & x_2 & + & x_3 \\
\text{subject to} & & & & & \\
& 2x_1 & + & 7.5x_2 & + & 3x_3 & \geq & 10000 \\
& 20x_1 & & & 5x_2 & + & 10x_3 & \geq & 30000 \\
& x_1, x_2, x_3 & & & & & \geq & 0 \quad .
\end{array}
$$

First, we convert this equation to the slack form. Doing so doesn't change the objective, but the constraints become

$$
\begin{array}{llllllll}
z & = & & - & x_1 & - & x_2 & - & x_3 \\
x_4 & = & -10000 & + & 2x_1 & + & 7.5x_2 & + & 3x_3 \\
x_5 & = & -30000 & + & 20x_1 & + & 5x_2 & + & 10x_3 \\
x_1, x_2, x_3, x_4, x_5 & \geq & 0 \quad .
\end{array}
$$

Also, since the objective is to minimize a given function, we'll change it over to maximizing the negative of that function. In particular maximize $-x_1 - x_2 - x_3$. Now, we note that the initial basic solution is not feasible, because it would leave x_4 and x_5 being negative. This means that finding an initial solution requires using the method of section 29.5. The auxiliary linear program in slack form is

$$
\begin{array}{llllllllll}
z & = & & - & x_0 \\
x_4 & = & -10000 & + & x_0 & + & 2x_1 & + & 7.5x_2 & + & 3x_3 \\
x_5 & = & -30000 & + & x_0 & + & 20x_1 & + & 5x_2 & + & 10x_3 \\
x_0, x_1, x_2, x_3, x_4, x_5 & \geq & 0 \quad .
\end{array}
$$

We choose x_0 as the entering variable and x_5 as the leaving variable, since it is the basic variable whose value in the basic solution is most negative. After pivoting, we have the slack form

$$
\begin{array}{llllllllll}
z & = & -30000 & + & 20x_1 & + & 5x_2 & + & 10x_3 & - & x_5 \\
x_0 & = & 30000 & - & 20x_1 & - & 5x_2 & - & 10x_3 & + & x_5 \\
x_4 & = & 20000 & - & 18x_1 & + & 2.5x_2 & - & 7x_3 & + & x_5 \\
x_0, x_1, x_2, x_3, x_4, x_5 & \geq & 0 \quad .
\end{array}
$$

The associated basic solution is feasible, so now we just need to repeatedly call PIVOT until we obtain an optimal solution to L_{aux} . We'll choose x_2 as our entering variable. This gives

$$
\begin{array}{llllllllll}
z & = & & - & x_0 \\
x_2 & = & 6000 & - & 0.2x_0 & - & 4x_1 & - & 2x_3 & + & 0.2x_5 \\
x_4 & = & 35000 & - & 0.5x_0 & - & 28x_1 & - & 12x_3 & + & 1.5x_5 \\
x_0, x_1, x_2, x_3, x_4, x_5 & \geq & 0 \quad .
\end{array}
$$

This slack form is the final solution to the auxiliary problem. Since this solution has $x_0 = 0$, we know that our initial problem was feasible. Furthermore, since $x_0 = 0$, we can just remove it from the set of constraints. We then restore the original objective function, with appropriate substitutions made to include only the nonbasic variables. This yields

$$
\begin{array}{llllllllll}
z & = & -6000 & + & 3x_1 & + & x_3 & - & 0.2x_5 \\
x_2 & = & 6000 & - & 4x_1 & - & 2x_3 & + & 0.2x_5 \\
x_4 & = & 35000 & - & 28x_1 & - & 12x_3 & + & 1.5x_5 \\
x_1, x_2, x_3, x_4, x_5 & \geq & 0 \quad .
\end{array}
$$

This slack form has a feasible basic solution, and we can return it to SIMPLEX. We choose x_1 as our entering variable. This gives

$$
\begin{aligned}
z &= -2250 &-& \tfrac{2}{7}x_3 &-& \tfrac{3}{28}x_4 &-& \tfrac{11}{280}x_5 \\
x_1 &= 1250 &-& \tfrac{3}{7}x_3 &-& \tfrac{1}{28}x_4 &+& \tfrac{15}{280}x_5 \\
x_2 &= 1000 &-& \tfrac{2}{7}x_3 &+& \tfrac{4}{28}x_4 &-& \tfrac{4}{280}x_5 \\
x_1, x_2, x_3, x_4, x_5 &\geq 0 & & & & & & .
\end{aligned}
$$

At this point, all coefficients in the objective function are negative, so the basic solution is an optimal solution. This solution is $(x_1, x_2, x_3) = (1250, 1000, 0)$.

29.3-8

> In the proof of Lemma 29.5, we argued that there are at most $\binom{m+n}{n}$ ways to choose a set B of basic variables. Give an example of a linear program in which there are strictly fewer than $\binom{m+n}{n}$ ways to choose the set B.

Consider the simple program

$$
\begin{aligned}
z &= &-& x_1 \\
x_2 &= 1 &-& x_1 & .
\end{aligned}
$$

In this case, we have $m = n = 1$, so $\binom{m+n}{n} = \binom{2}{1} = 2$, however, since the only coefficients of the objective function are negative, we can't make any other choices for basic variable. We must immediately terminate with the basic solution $(x_1, x_2) = (0, 1)$, which is optimal.

29.4 Duality

29.4-1

> Formulate the dual of the linear program given in Exercise 29.3-5.

By just transposing A, swapping b and c, and switching the maximization to a minimization, we want to minimize $20y_1 + 12y_2 + 16y_3$ subject to the constrain

$$
\begin{aligned}
y_1 + y_2 &\geq 18 \\
y_1 + y_3 &\geq 12.5 \\
y_1, y_2, y_3 &\geq 0
\end{aligned}
$$

29.4-2

> Suppose that we have a linear program that is not in standard form. We could produce the dual by first converting it to standard form, and then taking the dual. It would be more convenient, however, to be able to produce the dual directly. Explain how we can directly take the dual of an arbitrary linear program.

By working through each aspect of putting a general linear program into standard form, as outlined on page 852, we can show how to deal with transforming each into the dual individually. If the problem is a minimization instead of a maximization, replace c_j by $-c_j$ in (29.84). If there is a lack of nonnegativity constraint on x_j we duplicate the jth column of A, which corresponds to duplicating the jth row of A^T. If there is an equality constraint for b_i, we convert it to two inequalities by duplicating then negating the ith column of A^T, duplicating then negating the ith entry of b, and adding an extra y_i variable. We handle the greater-than-or-equal-to sign $\sum_{i=1}^{n} a_{ij}x_j \geq b_i$ by negating ith column of A^T and negating b_i. Then we solve the dual problem of minimizing $b^T y$ subject to $A^T y$ and $y \geq 0$.

29.4-3

> Write down the dual of the maximum-flow linear program, as given in lines (29.47)–(29.50) on page 860. Explain how to interpret this formulation as a minimum-cut problem.

First, we'll convert the linear program for maximum flow described in equation (29.47)-(29.50) into standard form. The objective function says that c is a vector indexed by a pair of vertices, and it is positive one if s is the first index and

negative one if s is the second index (zero if it is both). Next, we'll modify the constraints by switching the equalities over into inequalities to get

$$
\begin{array}{rcll}
f_{uv} & \leq & c(u,v) & \text{for each } u,v \in V \\
\sum_{u \in V} f_{vu} & \leq & \sum_{u \in V} f_{uv} & \text{for each } v \in V - \{s,t\} \\
\sum_{u \in V} f_{vu} & \geq & \sum_{u \in V} f_{uv} & \text{for each } v \in V - \{s,t\} \\
f_{uv} & \geq & 0 & \text{for each } u,v \in V
\end{array}
$$

Then, we'll convert all but the last set of the inequalities to be $\leq$ by multiplying the third line by -1.

$$
\begin{array}{rcll}
f_{uv} & \leq & c(u,v) & \text{for each } u,v \in V \\
\sum_{u \in V} f_{vu} & \leq & \sum_{u \in V} f_{uv} & \text{for each } v \in V - \{s,t\} \\
\sum_{u \in V} -f_{vu} & \leq & \sum_{u \in V} -f_{uv} & \text{for each } v \in V - \{s,t\} \\
f_{uv} & \geq & 0 & \text{for each } u,v \in V
\end{array}
$$

Finally, we'll bring all the variables over to the left to get

$$
\begin{array}{rcll}
f_{uv} & \leq & c(u,v) & \text{for each } u,v \in V \\
\sum_{u \in V} f_{vu} - \sum_{u \in V} f_{uv} & \leq & 0 & \text{for each } v \in V - \{s,t\} \\
\sum_{u \in V} -f_{vu} - \sum_{u \in V} -f_{uv} & \leq & 0 & \text{for each } v \in V - \{s,t\} \\
f_{uv} & \geq & 0 & \text{for each } u,v \in V
\end{array}
$$

Now, we can finally write down our A and b. A will be a $|V|^2 \times |V|^2 + 2|V| - 4$ matrix built from smaller matrices A_1 and A_2 which correspond to the three types of constraints that we have (of course, not counting the non-negativity constraints). We will let $g(u,v)$ be any bijective mapping from $V \times V$ to $[|V|^2]$. We'll also let h be any bijection from $V - \{s,t\}$ to $[|V| - 2]$.

$$
A = \begin{pmatrix} A_1 \\ A_2 \\ -A_2 \end{pmatrix},
$$

where A_1 is defined as having its row $g(u,v)$ be all zeroes except for having the value 1 at at the $g(u,v)$th entry. We define A_2 to have it's row $h(u)$ be equal to 1 at all columns j for which $j = g(v,u)$ for some v and equal to -1 at all columns j for which $j = g(u,v)$ for some v. Lastly, we mention that b is defined as having it's jth entry be equal to $c(u,v)$ if $j = g(u,v)$ and zero if $j > |V|^2$.

Now that we have placed the linear program in standard form, we can take its dual. We want to minimize $\sum_{i=1}^{|V|^2+2|V|-2} b_i y_i$ given the constraints that all the y values are non-negative, and $A^T y \geq c$.

29.4-4

> Write down the dual of the minimum-cost-flow linear program, as given in lines (29.51)–(29.52) on page 862. Explain how to interpret this problem in terms of graphs and flows.

First we need to put the linear programming problem into standard form, as follows:

$$
\begin{array}{rl}
\text{maximize} & \sum_{(u,v) \in E} -a(u,v) f_{uv} \\
\text{subject to} &
\end{array}
$$

$$
\begin{array}{rcll}
f_{uv} & \leq & c(u,v) & \text{for each } u,v \in V \\
\sum_{v \in V} f_{vu} - \sum_{v \in V} f_{uv} & \leq & 0 & \text{for each } u \in V - \{s,t\} \\
\sum_{v \in V} f_{uv} - \sum_{v \in V} f_{vu} & \leq & 0 & \text{for each } u \in V - \{s,t\} \\
\sum_{v \in V} f_{sv} - \sum_{v \in V} f_{vs} & \leq & d & \\
\sum_{v \in V} f_{vs} - \sum_{v \in V} f_{sv} & \leq & -d & \\
f_{uv} & \geq & 0 & .
\end{array}
$$

We now formulate the dual problem. Let the vertices be denoted $v_1, v_2, \ldots, v_n, s, t$ and the edges be $e_1, e_2, \ldots, e_k$. Then we have $b_i = c(e_i)$ for $1 \le i \le k$, $b_i = 0$ for $k + 1 \le i \le k + 2n$, $b_{k+2n+1} = d$, and $b_{k+2n+2} = -d$. We also have $c_i = -a(e_i)$ for $1 \le i \le k$. For notation, let $j.\,left$ denote the tail of edge e_j and $j.\,right$ denote the head. Let $\chi_s(e_j) = 1$ if e_j enters s, set it equal to -1 if e_j leaves s, and set it equal to 0 if e_j is not incident with s. The dual problem is:

$$\text{minimize} \quad \sum_{i=1}^{k} c(e_j)y_i + dy_{k+2n+1} - dy_{k+2n+2}$$
$$\text{subject to}$$

$$y_j + y_{k+e_j.right} - y_{k+j.left} - y_{k+n+e_j.right} + y_{k+n+e_j.left} - \chi_s(e_j)y_{k+2n+1} + \chi_s(e_j)y_{k+2n+2} \ge -a(e_j),$$

where j runs between 1 and k. There is one constraint equation for each edge e_j.

29.4-5

> Show that the dual of the dual of a linear program is the primal linear program.

Suppose that our original linear program is in standard form for some A, b, c. Then, the dual of this is to minimize $\sum_{i=1}^{m} b_i y_i$ subject to $A^T y \ge c$ This can be rewritten as wanting to maximize $\sum_{i=1}^{m} (-b_i)y_i$ subject to $(-A)^T y \le -c$. Since this is a standard form, we can take its dual easily, it is minimize $\sum_{j=1}^{n} (-c_j)x_j$ subject to $(-A)x \ge -b$. This is the same as minimizing $\sum_{j=1}^{n} c_j x_j$ subject to $Ax \le b$, which was the original linear program.

29.4-6

> Which result from Chapter 26 can be interpreted as weak duality for the maximum-flow problem?

Corollary 26.5 from Chapter 26 can be interpreted as weak duality.

29.5 The initial basic feasible solution

29.5-1

> Give detailed pseudocode to implement lines 5 and 14 of INITIALIZE-SIMPLEX.

For line 5, first let (N, B, A, b, c, v) be the result of calling PIVOT on L_{aux} using x_0 as the entering variable. Then repeatedly call PIVOT until an optimal solution to L_{aux} is obtained, and return this to (N, B, A, b, c, v). To remove x_0 from the constraints, set $a_{i,0} = 0$ for all $i \in B$, and set $N = N \setminus \{0\}$. To restore the original objective function of L, for each $j \in N$ and each $i \in B$, set $c_j = c_j - c_i a_{ij}$.

29.5-2

> Show that when the main loop of SIMPLEX is run by INITIALIZE-SIMPLEX, it can never return "unbounded."

In order to enter line 10 of INITIALIZE-SIMPLEX and begin iterating the main loop of SIMPLEX, we must have recovered a basic solution which is feasible for L_{aux}. Since $x_0 \ge 0$ and the objective function is $-x_0$, the objective value associated to this solution (or any solution) must be negative. Since the goal is to aximize, we have an upper bound of 0 on the objective value. By Lemma 29.2, SIMPLEX correctly determines whether or not the input linear program is unbounded. Since L_{aux} is not unbounded, this can never be returned by SIMPLEX.

29.5-3

> Suppose that we are given a linear program L in standard form, and suppose that for both L and the dual of L, the basic solutions associated with the initial slack forms are feasible. Show that the optimal objective value of L is 0.

Since it is in standard form, the objective function has no constant term, it is entirely given by $\sum_{i=1}^{n} c_i x_i$, which is going to be zero for any basic solution. The same thing goes for its dual. Since there is some solution which has the objective function achieve the same value both for the dual and the primal, by the corollary to the weak duality theorem, that common value must be the optimal value of the objective function.

29.5-4

> Suppose that we allow strict inequalities in a linear program. Show that in this case, the fundamental theorem of linear programming does not hold.

Consider the linear program in which we wish to maximize x_1 subject to the constraint $x_1 < 1$ and $x_1 \geq 0$. This has no optimal solution, but it is clearly bounded and has feasible solutions. Thus, the Fundamental theorem of linear programming does not hold in the case of strict inequalities.

29.5-5

> Solve the following linear program using SIMPLEX:

$$
\begin{array}{rlrlrl}
\text{maxmize} & \quad x_1 & + & 3x_2 & & \\
\text{subject to} & & & & & \\
& x_1 & - & x_2 & \leq & 8 \\
& -x_1 & - & x_2 & \leq & -3 \\
& -x_1 & + & 4x_2 & \leq & 2 \\
& x_1, x_2 & & & \geq & 0 \quad.
\end{array}
$$

The initial basic solution isn't feasible, so we will need to form the auxiliary linear program,

$$
\begin{array}{rlllllll}
\text{maxmize} & -x_0 & & & & & & \\
\text{subject to} & & & & & & & \\
& -x_0 & + & x_1 & - & x_2 & \leq & 8 \\
& -x_0 & - & x_1 & - & x_2 & \leq & -3 \\
& -x_0 & - & x_1 & + & 4x_2 & \leq & 2 \\
& x_0, x_1, x_2 & & & & & \geq & 0 \quad.
\end{array}
$$

Writing this linear program in slack form,

$$
\begin{array}{rclrlrlrl}
z & = & & - & x_0 & & & & \\
x_3 & = & 8 & + & x_0 & - & x_1 & + & x_2 \\
x_4 & = & -3 & + & x_0 & + & x_1 & + & x_2 \\
x_5 & = & 2 & + & x_0 & + & x_1 & - & 4x_2 \\
x_0, x_1, x_2, x_3, x_4, x_5 & \geq & 0 & . & & & & &
\end{array}
$$

Next we make one call to PIVOT where x_0 is the entering variable and x_4 is the leaving variable.

$$
\begin{array}{rclrlrlrl}
z & = & -3 & + & x_1 & + & x_2 & - & x_4 \\
x_0 & = & 3 & - & x_1 & - & x_2 & + & x_4 \\
x_3 & = & 11 & - & 2x_1 & & & + & x_4 \\
x_5 & = & 5 & & & - & 5x_2 & + & x_4 \\
x_0, x_1, x_2, x_3, x_4, x_5 & \geq & 0 & . & & & & &
\end{array}
$$

The basic solution is feasible, so we repeatedly call PIVOT to get the optimal solution to L_{aux}. We'll choose x_1 to be our entering variable and x_0 to be the leaving variable. This gives

$$
\begin{array}{rclrlrlrl}
z & = & & & -x_0 & & & & \\
x_1 & = & 3 & - & x_0 & - & x_2 & + & x_4 \\
x_3 & = & 5 & + & 2x_0 & + & 2x_2 & - & x_4 \\
x_5 & = & 5 & & & - & 5x_2 & + & x_4 \\
x_0, x_1, x_2, x_3, x_4, x_5 & \geq & 0 & . & & & & &
\end{array}
$$

The basic solution is now optimal for L_{aux}, so we return this slack form to SIMPLEX, set $x_0 = 0$, and update the objective function which yields

$$
\begin{aligned}
z &= 3 &+& 2x_2 &+& x_4 \\
x_1 &= 3 &-& x_2 &+& x_4 \\
x_3 &= 5 &+& 2x_2 &-& x_4 \\
x_5 &= 5 &-& 5x_2 &+& x_4 \\
x_1, x_2, x_3, x_4, x_5 &\geq 0 &&&&.
\end{aligned}
$$

We'll choose x_2 as our entering variable, which makes x_5 our leaving variable. PIVOT then gives,

$$
\begin{aligned}
z &= 5 &+& (7/5)x_4 &-& (2/5)x_5 \\
x_1 &= 2 &+& (1/5)x_4 &+& (1/5)x_5 \\
x_2 &= 1 &+& (4/5)x_4 &-& (1/5)x_5 \\
x_3 &= 7 &-& (3/5)x_4 &-& (2/5)x_5 \\
x_1, x_2, x_3, x_4, x_5 &\geq 0 &&&&.
\end{aligned}
$$

We'll choose x_4 as our entering variable, which makes x_3 our leaving variable. PIVOT then gives,

$$
\begin{aligned}
z &= (64/3) &-& (7/3)x_3 &-& (4/3)x_5 \\
x_1 &= (34/3) &-& (4/3)x_3 &-& (1/3)x_5 \\
x_2 &= (10/3) &-& (1/3)x_3 &-& (1/3)x_5 \\
x_4 &= (35/3) &-& (5/3)x_3 &-& (2/3)x_5 \\
x_1, x_2, x_3, x_4, x_5 &\geq 0 &&&&.
\end{aligned}
$$

Now all coefficients in the objective function are negative, so the basic solution is the optimal solution. It is $(x_1, x_2) = (34/3, 10/3)$.

29.5-6

Solve the following linear program using SIMPLEX:

$$
\begin{aligned}
\text{maxmize} \quad & x_1 - 2x_2 \\
\text{subject to} \\
& x_1 + 2x_2 \leq 4 \\
& -2x_1 - 6x_2 \leq -12 \\
& x_2 \leq 1 \\
& x_1, x_2 \geq 0 .
\end{aligned}
$$

The initial basic solution isn't feasible, so we will need to form the auxiliary linear program,

$$
\begin{aligned}
\text{maxmize} \quad & -x_0 \\
\text{subject to} \\
& -x_0 + x_1 + 2x_2 \leq 4 \\
& -x_0 - 2x_1 - 6x_2 \leq -12 \\
& -x_0 + x_2 \leq 1 \\
& x_0, x_1, x_2 \geq 0 .
\end{aligned}
$$

Writing this linear program in slack form,

$$
\begin{aligned}
z &= &-& x_0 \\
x_3 &= 4 &+& x_0 &-& x_1 &-& 2x_2 \\
x_4 &= -12 &+& x_0 &+& 2x_1 &+& 6x_2 \\
x_5 &= 1 &+& x_0 &&&-& x_2 \\
x_0, x_1, x_2, x_3, x_4, x_5 &\geq 0 &&.
\end{aligned}
$$

Next we make one call to PIVOT where x_0 is the entering variable and x_4 is the leaving variable.

$$
\begin{array}{rcrrrrrrrr}
z & = & -12 & + & 2x_1 & + & 6x_2 & - & x_4 \\
x_0 & = & 12 & - & 2x_1 & - & 6x_2 & + & x_4 \\
x_3 & = & 16 & - & 3x_1 & - & 8x_2 & + & x_4 \\
x_5 & = & 13 & - & 2x_1 & - & 8x_2 & + & x_4 \\
x_0, x_1, x_2, x_3, x_4, x_5 & \geq & 0 & . &
\end{array}
$$

The basic solution is $(x_0, x_1, x_2, x_3, x_4, x_5) = (12, 0, 0, 16, 0, 13)$ which is feasible for the auxiliary program. Now we need to run SIMPLEX to find the optimal objective value to L_{aux}. Let x_1 be our next entering variable. It is most constrained by x_3, which will be our leaving variable. After PIVOT, the new linear program is

$$
\begin{array}{rcrrrrrrrr}
z & = & -(4/3) & + & (2/3)x_2 & - & (2/3)x_3 & + & (1/3)x_4 \\
x_0 & = & (4/3) & - & (2/3)x_2 & + & (2/3)x_3 & + & (1/3)x_4 \\
x_1 & = & (16/3) & - & (8/3)x_2 & - & (1/3)x_3 & + & (1/3)x_4 \\
x_5 & = & (7/3) & - & (8/3)x_2 & + & (2/3)x_3 & + & (1/3)x_4 \\
x_0, x_1, x_2, x_3, x_4, x_5 & \geq & 0 & . &
\end{array}
$$

Every coefficient in the objective function is negative, so we take the basic solution
$(x_0, x_1, x_2, x_3, x_4, x_5) = (4/3, 16/3, 0, 0, 0, 7/3)$ which is also optimal. Since $x_0 \neq 0$, the original linear program must be unfeasible.

29.5-7

> Solve the following linear program using SIMPLEX:
>
> $$
> \begin{array}{rrrrrr}
> \text{maxmize} & x_1 & + & 3x_2 & & \\
> \text{subject to} & & & & & \\
> & -x_1 & + & x_2 & \leq & -1 \\
> & -x_1 & - & x_2 & \leq & -3 \\
> & -x_1 & + & 4x_2 & \leq & 2 \\
> & x_1, x_2 & & & \geq & 0 & .
> \end{array}
> $$

The initial basic solution isn't feasible, so we will need to form the auxiliary linear program,

$$
\begin{array}{rrrrrrrr}
\text{maxmize} & -x_0 & & & & & & \\
\text{subject to} & & & & & & & \\
& -x_0 & - & x_1 & + & x_2 & \leq & -1 \\
& -x_0 & - & x_1 & - & x_2 & \leq & -3 \\
& -x_0 & - & x_1 & + & 4x_2 & \leq & 2 \\
& x_0, x_1, x_2 & & & & & \geq & 0 & .
\end{array}
$$

Writing this linear program in slack form,

$$
\begin{array}{rcrrrrrrrr}
z & = & & & - & x_0 & & & & \\
x_3 & = & -1 & + & x_0 & + & x_1 & - & x_2 \\
x_4 & = & -3 & + & x_0 & + & x_1 & + & x_2 \\
x_5 & = & 2 & + & x_0 & + & x_1 & - & 4x_2 \\
x_0, x_1, x_2, x_3, x_4, x_5 & \geq & 0 & . &
\end{array}
$$

Next we make one call to PIVOT where x_0 is the entering variable and x_4 is the leaving variable.

$$
\begin{array}{rcrrrrrrrr}
z & = & -3 & + & x_1 & + & x_2 & - & x_4 \\
x_0 & = & 3 & - & x_1 & - & x_2 & + & x_4 \\
x_3 & = & 2 & & & - & 2x_2 & + & x_4 \\
x_5 & = & 5 & & & - & 5x_2 & + & x_4 \\
x_0, x_1, x_2, x_3, x_4, x_5 & \geq & 0 & . &
\end{array}
$$

Let x_1 be our entering variable. Then x_0 is our leaving variable, and we have

$$
\begin{array}{rclcccccccc}
z & = & & & - & x_0 & & & & & \\
x_1 & = & 3 & & - & x_0 & - & x_2 & + & x_4 \\
x_3 & = & 2 & & & & - & 2x_2 & + & x_4 \\
x_5 & = & 5 & & & & - & 5x_2 & + & x_4 \\
x_0, x_1, x_2, x_3, x_4, x_5 & \geq & 0 & . & & & & & &
\end{array}
$$

The basic solution is feasible, and optimal for L_{aux}, so we return this and run SIMPLEX. Updating the objective function and setting $x_0 = 0$ gives

$$
\begin{array}{rclcccccc}
z & = & 3 & + & 2x_2 & + & x_4 \\
x_1 & = & 3 & - & x_2 & + & x_4 \\
x_3 & = & 2 & - & 2x_2 & + & x_4 \\
x_5 & = & 5 & - & 5x_2 & + & x_4 \\
x_0, x_1, x_2, x_3, x_4, x_5 & \geq & 0 & . & & &
\end{array}
$$

We'll choose x_2 as our entering variable, which makes x_3 our leaving variable. This gives

$$
\begin{array}{rclcccccc}
z & = & 5 & - & & x_3 & + & 2x_4 \\
x_1 & = & 2 & + & (1/2)x_3 & + & (1/2)x_4 \\
x_2 & = & 1 & - & (1/2)x_3 & + & (1/2)x_4 \\
x_5 & = & & & (5/2)x_3 & - & (3/2)x_4 \\
x_0, x_1, x_2, x_3, x_4, x_5 & \geq & 0 & . & & &
\end{array}
$$

Next we use x_4 as our entering variable, which makes x_5 our leaving variable. This gives

$$
\begin{array}{rclcccccc}
z & = & 5 & + & (7/3)x_3 & - & (4/3)x_5 \\
x_1 & = & 2 & + & (4/3)x_3 & - & (1/3)x_5 \\
x_2 & = & 1 & + & (1/3)x_3 & - & (1/3)x_5 \\
x_4 & = & & & (5/3)x_3 & - & (2/3)x_5 \\
x_0, x_1, x_2, x_3, x_4, x_5 & \geq & 0 & . & & &
\end{array}
$$

Finally, we would like to choose x_3 as our entering variable, but every coefficient on x_3 is positive, so SIMPLEX returns that the linear program is unbounded.

29.5-8

> Solve the linear program given in (29.6)–(29.10).

We first put the linear program in standard form,

$$
\begin{array}{llcccccccccc}
\text{maxmize} & x_1 & + & x_2 & + & x_3 & + & x_4 \\
\text{subject to} & & & & & & & \\
& 2x_1 & - & 8x_2 & & & - & 10x_4 & \leq & -50 \\
& -5x_1 & - & 2x_2 & & & & & \leq & -100 \\
& -3x_1 & + & 5x_2 & - & 10x_3 & + & 2x_4 & \leq & -25 \\
& x_1, x_2, x_3, x_4 & & & & & & & \geq & 0 & .
\end{array}
$$

The initial basic solution isn't feasible, so we will need to form the auxiliary linear program.

$$
\begin{array}{rclccccccccccc}
z & = & & & - & x_0 & & & & & & & \\
x_5 & = & -50 & + & x_0 & - & 2x_1 & + & 8x_2 & & & + & 10x_4 \\
x_6 & = & -100 & + & x_0 & + & 5x_1 & + & 2x_2 & & & & \\
x_7 & = & -25 & + & x_0 & + & 3x_1 & - & 5x_2 & + & 10x_3 & - & 2x_4 \\
x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7 & \geq & 0 & . & & & & & & & & &
\end{array}
$$

The index of the minimum b_i is 2, so we take x_0 to be our entering variable and x_6 to be our leaving variable. The call to PIVOT on line 8 yields

$$
\begin{array}{rclrlrlrlrlrl}
z & = & -100 & + & 5x_1 & + & 2x_2 & & & & & - & x_6 \\
x_0 & = & 100 & - & 5x_1 & - & 2x_2 & & & & & + & x_6 \\
x_5 & = & 50 & - & 7x_1 & + & 8x_2 & & & + & 10x_4 & + & x_6 \\
x_7 & = & 75 & - & 2x_1 & - & 7x_2 & + & 10x_3 & - & 2x_4 & + & x_6 \\
\end{array}
$$
$$x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0 \ .$$

Next we'll take x_2 to be our entering variable and x_5 to be our leaving variable. The call to PIVOT yields

$$
\begin{array}{rclrlrlrlrl}
z & = & -225/2 & + & (27/4)x_1 & & & - & (10/4)x_4 & + & (1/4)x_5 & - & (5/4)x_6 \\
x_0 & = & 225/2 & - & (27/4)x_1 & & & + & (10/4)x_4 & - & (1/4)x_5 & + & (5/4)x_6 \\
x_2 & = & -50/8 & + & (7/8)x_1 & & & - & (10/8)x_4 & + & (1/8)x_5 & - & (1/8)x_6 \\
x_7 & = & 475/4 & - & (65/8)x_1 & + & 10x_3 & + & (54/8)x_4 & - & (7/8)x_5 & + & (15/8)x_6 \\
\end{array}
$$
$$x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0 \ .$$

The work gets rather messy, but INITIALIZE-SIMPLEX does eventually give a feasible solution to the linear program, and after running the simplex method we find that $(x_1, x_2, x_3, x_4) = (175/11, 225/22, 125/44, 0)$ is an optimal solution to the original linear programming problem.

29.5-9

Consider the following 1-variable linear program, which we call P:

$$
\begin{array}{rlrl}
\text{maximize} & tx & & \\
\text{subject to} & rx & \leq & s \\
& x & \geq & 0 \ ,
\end{array}
$$

where r, s, and t are arbitrary real numbers. Let D be the dual of P.

State for which values of r, s, and t you can assert that

1. Both P and D have optimal solutions with finite objective values.
2. P is feasible, but D is infeasible.
3. D is feasible, but P is infeasible.
4. Neither P nor D is feasible.

1. One option is that $r = 0, s \geq 0$ and $t \leq 0$. Suppose that $r > 0$, then, if we have that s is non-negative and t is non-positive, it will be as we want.
2. We will split into two cases based on r. If $r = 0$, then this is exactly when t is non-positive and s is non-negative. The other possible case is that r is negative, and t is positive. In which case, because r is negative, we can always get rx as small as we want so s doesn't matter, however, we can never make rx positive so it can never be $\geq t$.
3. Again, we split into two possible cases for r. If $r = 0$, then it is when t is nonnegative and s is non-positive. The other possible case is that r is positive, and s is negative. Since r is positive, rx will always be non-negative, so it cannot be $\leq s$. But since r is positive, we have that we can always make rx as big as we want, in particular, greater than t.
4. If we have that $r = 0$ and t is positive and s is negative. If r is nonzero, then we can always either make rx really big or really small depending on the sign of r, meaning that either the primal or the dual would be feasable.

Problem 29-1 Linear-inequality feasibility

Given a set of m linear inequalities on n variables $x_1, x_2, \ldots, x_n$, the *linear-inequality feasibility problem* asks whether there is a setting of the variables that simultaneously satisfies each of the inequalities.

a. Show that if we have an algorithm for linear programming, we can use it to solve a linear-inequality feasibility problem. The number of variables and constraints that you use in the linear-programming problem should be polynomial in n and m.

> **b.** Show that if we have an algorithm for the linear-inequality feasibility problem, we can use it to solve a linear-programming problem. The number of variables and linear inequalities that you use in the linear-inequality feasibility problem should be polynomial in n and m, the number of variables and constraints in the linear program.

a. We just let the linear inequalities that we need to satisfy be our set of constraints in the linear program. We let our function to maximize just be a constant. The solver for linear programs would fail to detect any feasible solution if the linear constraints were not feasible. If the linear programming solver returns any solution at all, we know that the linear constraints are feasible.

b. Suppose that we are trying to solve the linear program in standard form with some particular A, b, c. That is, we want to maximize $\sum_{j=1}^{n} c_j x_j$ subject to $Ax \le b$ and all entries of the x vector are non-negative. Now, consider the dual program, that is, we want to minimize $\sum_{i=1}^{m} b_i y_i$ subject to $A^T y \ge c$ and all the entries in the y vector are nonzero. We know by Corollary 29.9, if x and y are feasible solutions to their respective problems, then, if we have that their objective functions are equal, then, they are both optimal solutions.

We can force their objective functions to be equal. To do this, let c_k be some nonzero entry in the c vector. If there are no nonzero entries, then the function we are trying to optimize is just the zero function, and it is exactly a feasibility question, so we we would be done. Then, we add two linear inequalities to require $x_k = \frac{1}{c_k} \left(\sum_{i=1}^{m} b_i y_i - \sum_{j=1}^{n} c_j x_j \right)$.

This will require that whatever values the variables take, their objective functions will be equal. Lastly, we just throw these in with the inequalities we already had. So, the constraints will be:

$$Ax \le b$$
$$A^T y \ge c$$
$$x_k \le \frac{1}{c_k} \left(\sum_{i=1}^{m} b_i y_i - \sum_{j=1}^{n} c_j x_j \right)$$
$$x_k \ge \frac{1}{c_k} \left(\sum_{i=1}^{m} b_i y_i - \sum_{j=1}^{n} c_j x_j \right)$$
$$x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m \ge 0.$$

We have a number of variables equal to $n + m$ and a number of constraints equal to $2 + 2n + 2m$, so both are polynomial in n and m. Also, any assignment of variables which satisfy all of these constraints will be a feasible solution to both the problem and its dual that cause the respective objective functions to take the same value, and so, must be an optimal solution to both the original problem and its dual. This of course assumes that the linear inequality feasibility solver doesn't merely say that the inequalities are satisfiable, but actually returns a satisfying assignment.

Lastly, it is necessary to note that if there is some optimal solution x, then, we can obtain an optimal solution for the dual that makes the objective functions equal by theorem 29.10. This ensures that the two constraints we added to force the objectives of the primal and the dual to be equal don't cause us to change the optimal solution to the linear program.

Problem 29-2 Complementary slackness

> ***Complementary slackness*** describes a relationship between the values of primal variables and dual constraints and between the values of dual variables and primal constraints. Let $\overline{x}$ be a feasible solution to the primal linear program given in (29.16)–(29.18), and let $\overline{y}$ be a feasible solution to the dual linear program given in (29.83)–(29.85). Complementary slackness states that the following conditions are necessary and sufficient for $\overline{x}$ and $\overline{y}$ to be optimal:
>
> $$\sum_{i=1}^{m} a_{ij} \overline{y}_i = c_j \text{ or } \overline{x}_j = 0 \text{ for } j = 1, 2, \ldots, n$$
>
> and
>
> $$\sum_{j=1}^{m} a_{ij} \overline{x}_j = b_i \text{ or } \overline{y}_i = 0 \text{ for } j = 1, 2, \ldots, m.$$

a. Verify that complementary slackness holds for the linear program in lines (29.53)–(29.57).

b. Prove that complementary slackness holds for any primal linear program and its corresponding dual.

c. Prove that a feasible solution $\bar{x}$ to a primal linear program given in lines (29.16)–(29.18) is optimal if and only if there exist values $\bar{y} = (\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_m)$ such that

1. $\bar{y}$ is a feasible solution to the dual linear program given in (29.83)–(29.85),
2. $\sum_{i=1}^{m} a_{ij}\bar{y}_i = c_j$ for all j such that $\bar{x}_j > 0$, and
3. $\bar{y}_i = 0$ for all i such that $\sum_{j=1}^{n} a_{ij}\bar{x}_j < b_i$.

a. An optimal solution to the LP program given in (29.53)-(29.57) is $(x_1, x_2, x_3) = (8, 4, 0)$. An optimal solution to the dual is $(y_1, y_2, y_3) = (0, 1/6, 2/3)$. It is then straightforward to verify that the equations hold.

b. First suppose that complementary slackness holds. Then the optimal objective value of the primal problem is, if it exists,

$$\sum_{k=1}^{n} c_k x_k = \sum_{k=1}^{n} \sum_{i=1}^{m} a_{ik} y_i x_k$$
$$= \sum_{i=1}^{m} \sum_{k=1}^{n} a_{ik} x_k y_i$$
$$= \sum_{i=1}^{m} b_i y_i,$$

which is precisely the optimal objective value of the dual problem. If any x_j is 0, then those terms drop out of them sum, so we can safely replace c_k by whatever we like in those terms. Since the objective values are equal, they must be optimal. An identical argument shows that if an optimal solution exists for the dual problem then any feasible solution for the primal problem which satisfies the second equality of complementary slackness must also be optimal.

Now suppose that x and y are optimal solutions, but that complementary slackness fails. In other words, there exists some j such that $x_j \neq 0$ but $\sum_{i=1}^{m} a_{ij} y_i > c_j$, or there exists some i such that $y_i \neq 0$ but $\sum_{j=1}^{n} a_{ij} x_j < b_i$. In the first case we have

$$\sum_{k=1}^{n} c_k x_k < \sum_{k=1}^{n} \sum_{i=1}^{m} a_{ik} y_i x_k$$
$$= \sum_{i=1}^{m} \sum_{k=1}^{n} a_{ik} x_k y_i$$
$$= \sum_{i=1}^{m} b_i y_i.$$

This implies that the optimal objective value of the primal solution is strictly less than the optimal value of the dual solution, a contradiction. The argument for the second case is identical. Thus, x and y are optimal solutions if and only if complementary slackness holds.

c. This follows immediately from part (b). If x is feasible and y satisfies conditions 1, 2, and 3, then complementary slackness holds, so x and y are optimal. On the other hand, if x is optimal, then the dual linear program must have an optimal solution y as well, according to Theorem 29.10. Optimal solutions are feasible, and by part (b), x and y satisfy complementary slackness. Thus, conditions 1, 2, and 3 hold.

Problem 29-3 Integer linear programming

An ***integer linear-programming problem*** is a linear-programming problem with the additional constraint that the variables x must take on integral values. Exercise 34.5-3 shows that just determining whether an integer linear program has a feasible solution is NP-hard, which means that there is no known polynomial-time algorithm for this problem.

> **a.** Show that weak duality (Lemma 29.8) holds for an integer linear program.
>
> **b.** Show that duality (Theorem 29.10) does not always hold for an integer linear program.
>
> **c.** Given a primal linear program in standard form, let us define P to be the optimal objective value for the primal linear program, D to be the optimal objective value for its dual, IP to be the optimal objective value for the integer version of the primal (that is, the primal with the added constraint that the variables take on integer values), and ID to be the optimal objective value for the integer version of the dual. Assuming that both the primal integer program and the dual integer program are feasible and bounded, show that
>
> $$IP \le P = D \le ID.$$

a. The proof for weak duality goes through identically. Nowhere in it does it use the integrality of the solutions.

b. Consider the linear program given in standard form by $A = (1)$, $b = (\frac{1}{2})$ and $c = (2)$. The highest we can get this is 0 since that's that only value that x can be. Now, consider the dual to this, that is, we are trying to minimize $\frac{x}{2}$ subject to the constraint that $x \ge 2$. This will be minimized when $x = 2$, so, the smallest solution we can get is 1.

Since we have just exhibited an example of a linear program that has a different optimal solution as it's dual, the duality theorem does not hold for integer linear program.

c. The first inequality comes from looking at the fact that by adding the restriction that the solution must be integer valued, we obtain a set of feasible solutions that is a subset of the feasible solutions of the original primal linear program. Since, to get IP, we are taking the max over a subset of the things we are taking a max over to get P, we must get a number that is no larger. The third inequality is similar, except since we are taking min over a subset, the inequality goes the other way. The middle equality is given by Theorem 29.10.

Problem 29-4 Farkas'ss lemma

> Let A be an $m \times n$ matrix and c be an n-vector. Then Farkas's lemma states that exactly one of the systems
>
> $$Ax \le 0,$$
> $$c^T x > 0$$
>
> and
>
> $$A^T y = c,$$
> $$y \le 0$$
>
> is solvable, where x is an n-vector and y is an m-vector. Prove Farkas's lemma.

Suppose that both systems are solvable, let x denote a solution to the first system, and y denote a solution to the second. Taking transposes we have $x^T A^T \le 0^T$. Right multiplying by y gives $x^T c = x^T A^T y \le 0^T$, which is a contradiction to the fact that $c^T x > 0$. Thus, both systems cannot be simultaneously solved. Now suppose that the second system fails. Consider the following linear program:

$$\text{maximize } 0x \text{ subject to } A^T y = c \text{ and } y \ge 0,$$

and its corresponding dual program

$$\text{minimize } -c^T x \text{ subject to } Ax \le 0.$$

Since the second system fails, the primal is infeasible. However, the dual is always feasible by taking $x = 0$. If there were a finite solution to the dual, then duality says there would also be a finite solution to the primal. Thus, the dual must be unbounded. Thus, there must exist a vector x which makes $-c^T x$ arbitrarily small, implying that there exist vectors x for which $c^T x$ is strictly greater than 0. Thus, there is always at least one solution.

Problem 29-5 Minimum-cost circulation

In this problem, we consider a variant of the minimum-cost-flow problem from Section 29.2 in which we are not given a demand, a source, or a sink. Instead, we are given, as before, a flow network and edge costs $a(u, v)$ flow is feasible if it satisfies the capacity constraint on every edge and flow conservation at *every* vertex. The goal is to find, among all feasible flows, the one of minimum cost. We call this problem the ***minimum-cost-circulation problem***.

a. Formulate the minimum-cost-circulation problem as a linear program.

b. Suppose that for all edges $(u, v) \in E$, we have $a(u, v) > 0$. Characterize an optimal solution to the minimum-cost-circulation problem.

c. Formulate the maximum-flow problem as a minimum-cost-circulation problem linear program. That is given a maximum-flow problem instance $G = (V, E)$ with source s, sink t and edge capacities c, create a minimum-cost-circulation problem by giving a (possibly different) network $G' = (V', E')$ with edge capacities c' and edge costs a' such that you can discern a solution to the maximum-flow problem from a solution to the minimum-cost-circulation problem.

d. Formulate the single-source shortest-path problem as a minimum-cost-circulation problem linear program.

a. This is exactly the linear program given in equations (29.51)-(29.52) except that the equation on the third line of the constraints should be removed, and for the equation on the second line of the constraints, u should be selected from all of V instead of from $V \setminus \{s, t\}$.

b. If $a(u, v) > 0$ for every pair of vertices, then, there is no point in sending any flow at all. So, an optimal solution is just to have no flow. This obviously satisfies the capacity constraints, it also satisfies the conservation constraints because the flow into and out of each vertex is zero.

c. We assume that the edge (t, s) is not in E because that would be a silly edge to have for a maximum flow from s to t. If it is there, remove it and it won't decrease the maximum flow. Let $V' = V$ and $E' = E \cup \{(t, s)\}$. For the edges of E' that are in E, let the capacity be as it is in E and let the cost be 0. For the other edge, we set $c(t, s) = \infty$ and $a(t, s) = -1$. Then, if we have any circulation in G', it will be trying to get as much flow to go across the edge (t, s) in order to drive the objective function lower, the other flows will have no affect on the objective function. Then, by Kirchhoff's current law (a.k.a. common sense), the amount going across the edge (t, s) is the same as the total flow in the rest of the network from s to t. This means that maximizing the flow across the edge (t, s) is also maximizing the flow from s to t. So, all we need to do to recover the maximum flow for the original network is to keep the same flow values, but throw away the edge (t, s).

d. Suppose that s is the vertex that we are computing shortest distance from. Then, we make the circulation network by first starting with the original graph, giving each edge a cost of whatever it was before and infinite capacity. Then, we place an edge going from every vertex that isn't s to s that has a capacity of 1 and a cost of $-|E|$ times the largest cost appearing among all the edge costs already in the graph. Giving it such a negative cost ensures that placing other flow through the network in order to get a unit of flow across it will cause the total cost to decrease. Then, to recover the shortest path for any vertex, start at that vertex and then go to any vertex that is sending a unit of flow to it. Repeat this until you've reached s.