Problem 1

Look at the tableau right before the pivot. Without loss of generality we may assume that the last \( m \) columns form the basis (we can always permute the columns, which amounts to changing variable indices). Assume that some column \( A_j = (a_{1,j}, \ldots, a_{m,j}) \leq 0 \) is profitable, i.e. its relative cost \( \bar{c}_j \) is negative.

<table>
<thead>
<tr>
<th>(-c \cdot x)</th>
<th>(\cdots)</th>
<th>(\bar{c}_j)</th>
<th>(\cdots)</th>
<th>(0)</th>
<th>(\cdots)</th>
<th>(0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_{n-m+1})</td>
<td>(a_{1,j})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(I_n)</td>
</tr>
<tr>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(\vdots)</td>
</tr>
<tr>
<td>(x_n)</td>
<td>(a_{m,j})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(\vdots)</td>
</tr>
</tbody>
</table>

The current (feasible) solution is \( x = (0, \ldots, 0, x_{n-m+1}, \ldots, x_n) \). The key observation is that for all \( \theta \geq 0 \) the vector

\[
x_\theta = x + \theta e_j + (0, \ldots, 0, -\theta a_{1,j}, \ldots, -\theta a_{m,j}),
\]

where \( e_j \) is the \( j \)th base vector in \( \mathbb{R}^n \), will also be a solution, and since \( \theta, x_{i,j} - \theta a_{i,j} \geq 0 \) for all \( i \), \( x_\theta \) is feasible. Finally, note that the cost of the new solution is

\[
c \cdot x_\theta = c \cdot x + \theta \bar{c}_j,
\]

and since \( \bar{c}_j < 0 \), it can be made arbitrarily small.

Problem 2

Recall that a pivot of the simplex algorithm will try to move the feasible point only in a direction such that a unit change of its position results in a non-zero decrease of the cost. Formally, the change in the cost function that is the result of moving that feasible point equals \( \theta \bar{c}_j \), where \( \theta > 0 \) if and only if the feasible point was moved a positive distance. Hence the answer is no – a pivot of the simplex algorithm cannot move the feasible point a positive distance without changing the cost.

Problem 3

Suppose that \( j \)th column enters the basis, \( k \)th column leaves it, and we pivot in \( i \)th row, as illustrated by the following tableau:

<table>
<thead>
<tr>
<th>(-c \cdot x)</th>
<th>(\cdots)</th>
<th>(\bar{c}_j)</th>
<th>(\cdots)</th>
<th>(0)</th>
<th>(\cdots)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\ast)</td>
<td>(\ast)</td>
<td>(0)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\ast)</td>
<td>(a_{i,j})</td>
<td>(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\ast)</td>
<td>(\ast)</td>
<td>(0)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note that in order to do the pivoting, \( a_{i,j} > 0 \) and \( \bar{c}_j < 0 \). During the change of basis we perform row operations to transform the \( j \)th column into the form \((0, \ldots, 1, \ldots, 0)\) and make \( \bar{c}_j = 0 \). These row operations alter the value of \( \bar{c}_k \), which was initially 0, and now becomes \( \bar{c}_k = -\bar{c}_j / a_{i,j} > 0 \), and therefore \( k \)th column cannot be considered as a pivot column in the very next step of the simples algorithm, since its relative cost is now positive.
Problem 4
Consider the following LP:
\[
A = \begin{pmatrix}
1 & 1 & 1 & 0 & 0 \\
-1 & 1 & 0 & 1 & 0 \\
1 & -1 & 0 & 0 & 1
\end{pmatrix}, \quad b = (1, 0, 0), \quad c = (-1, -1, 0, 0, 0).
\]
Suppose the algorithm is at the b.f.s. \(x_0 = (0, 0, 1, 0, 0)\), and the current basis consists of columns \(A_3, A_4, A_5\). We can try to pivot \(A_1\) or \(A_2\) into the basis. Note that
\[
A_1 = 1 \cdot A_3 + (-1) \cdot A_4 + 1 \cdot A_5
\]
\[
A_2 = 1 \cdot A_3 + 1 \cdot A_4 + (-1) \cdot A_5
\]
\[
b = 1 \cdot A_3 + 0 \cdot A_4 + 0 \cdot A_5,
\]
so in both cases the algorithm will compute \(\theta = \min \left\{ \frac{1}{4}, \frac{0}{7} \right\} = 0\). Nevertheless \(x_0\) is not an optimal solution, since it is not hard to see that if we set \(x_1 = (1, 1, 0, 0, 0)\), then
\[
c \cdot x_1 = -2 < 0 = c \cdot x_0.
\]

Problem 5
Suppose that only one basic variable is equal to 0. If the algorithm is going to cycle, then at each step in that cycle the cost must remain the same, and so the computed \(\theta\) must be 0. Hence the row in which we pivot has to be the same unique row corresponding to the only zero basic variable.
Suppose that cycling started with \(k\)th column leaving the basis. Repeating the argument from the solution of Problem 3, immediately after a column leaves the basis, its relative cost becomes positive. Hence \(\bar{c}_k > 0\). Note that before this column can reenter the basis, its relative cost \(\bar{c}_k\) must be negative again. Since every time we pivot in the same row, say \(i\)th row, \(\bar{c}_k\) increases by \(-a_{i,k} \cdot \bar{c}_j / a_{i,j} > 0\) (column \(j\) enters the basis; \(a_{i,j}, a_{i,k} > 0\), \(\bar{c}_j < 0\)) and the entry \(a_{i,k}\) gets replaced by \(a_{i,k} / a_{i,j} > 0\). Hence unless we start pivoting in another row (which contradicts our assumption), \(\bar{c}_k\) will remain positive and column \(k\) will never reenter the basis.

Problem 6
First note that the cone of an empty set is \(\{0\}\), hence in the special case we are investigating, Farkas’ Lemma can be stated as follows:
\[
\forall y \left[ \left( \forall a \in \emptyset \right) y \cdot a \geq 0 \Rightarrow y \cdot c \geq 0 \right] \iff c = 0,
\]
and since any formula quantified over an empty set is vacuously true, it is equivalent to
\[
\forall y \ y \cdot c \geq 0 \iff c = 0.
\]
The implication from right to left is trivial – for every \(y\), \(y \cdot 0 = 0 \geq 0\). For the other direction, if \(y \cdot c \geq 0\) for all \(y\), then in particular \(-\|c\|^2 = (-c) \cdot c \geq 0\), hence \(\|c\|^2 = 0\), and so \(c = 0\).