CS 373: Combinatorial Algorithms, Spring 2002
This is not homework 8 - DO NOT SUBMIT

The following is what hw8 would have been, if we had enough time to check it (we don’t). This homework is an exercise - solve it if you want, don’t solve it if you don’t feel like it. Whatever you do don’t submit it.

However, there is a non-zero probability that a question from this homework would appear in the final exam. Ignore this exercise at your own risk.

No solutions for this exercise would be posted.

Problems

1. [20 Points] Provide detailed solutions for the following problems, showing each pivoting stage separately.

(a) [5 Points]
maximize $6x_1 + 8x_2 + 5x_3 + 9x_4$
subject to $2x_1 + x_2 + x_3 + 3x_4 \leq 5$
$x_1 + 3x_2 + x_3 + 2x_4 \leq 3$
$x_1, x_2, x_3, x_4 \geq 0$.

(b) [5 Points]
maximize $2x_1 + x_2$
subject to $2x_1 + x_2 \leq 4$
$2x_1 + 3x_2 \leq 3$
$4x_1 + x_2 \leq 5$
$x_1 + 5x_2 \leq 1$
$x_1, x_2 \geq 0$.

(c) [5 Points]
maximize $6x_1 + 8x_2 + 5x_3 + 9x_4$
subject to $x_1 + x_2 + x_3 + x_4 = 1$
$x_1, x_2, x_3, x_4 \geq 0$.

(d) [5 Points]
minimize $x_{12} + 8x_{13} + 9x_{14} + 2x_{23} + 7x_{24} + 3x_{34}$
subject to $x_{12} + x_{13} + x_{14} \geq 1$
$-x_{12} + x_{23} + x_{24} = 0$
$-x_{13} - x_{23} + x_{34} = 0$
$x_{14} + x_{24} + x_{34} \leq 1$
$x_{12}, x_{13}, \ldots, x_{34} \geq 0$.

2. [5 Points] A steel company must decide how to allocate next week’s time on a rolling mill, which is a machine that takes unfinished slabs of steel as input and produce either of two
semi-finished products: bands and coils. The mill’s two products come off the rolling line at different rates:

- Bands 200 tons/hr
- Coils 140 tons/hr.

They also produce different profits:

- Bands $25/ton
- Coils $30/ton.

Based on current booked orders, the following upper bounds are placed on the amount of each product to produce:

- Bands 6000 tons
- Coils 4000 tons.

Given that there are 40 hours of production time available this week, the problem is to decide how many tons of bands and how many tons of coils should be produced to yield the greatest profit. Formulate this problem as a linear programming problem. Can you solve this problem by inspection?

3. **[5 Points]** A small airline, Ivy Air, flies between three cities: Ithaca (a small town in upstate New York), Newark (an eyesore in beautiful New Jersey), and Boston (a yuppie town in Massachusetts). They offer several flights but, for this problem, let us focus on the Friday afternoon flight that departs from Ithaca, stops in Newark, and continues to Boston. There are three types of passengers:

   (a) Those traveling from Ithaca to Newark (god only knows why).
   (b) Those traveling from Newark to Boston (a very good idea).
   (c) Those traveling from Ithaca to Boston (it depends on who you know).

The aircraft is a small commuter plane that seats 30 passengers. The airline offers three fare classes:

   (a) Y class: full coach.
   (b) B class: nonrefundable.
   (c) M class: nonrefundable, 3-week advanced purchase.

Ticket prices, which are largely determined by external influences (i.e., competitors), have been set and advertised as follows:

<table>
<thead>
<tr>
<th></th>
<th>Ithaca-Newark</th>
<th>Newark-Boston</th>
<th>Ithaca-Boston</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>300</td>
<td>160</td>
<td>360</td>
</tr>
<tr>
<td>B</td>
<td>220</td>
<td>130</td>
<td>280</td>
</tr>
<tr>
<td>M</td>
<td>100</td>
<td>80</td>
<td>140</td>
</tr>
</tbody>
</table>
Based on past experience, demand forecasters at Ivy Air have determined the following upper bounds on the number of potential customers in each of the 9 possible origin-destination/fare-class combinations:

<table>
<thead>
<tr>
<th></th>
<th>Ithaca-Newark</th>
<th>Newark-Boston</th>
<th>Ithaca-Boston</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ithaca-Newark</td>
<td>4</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>Newark-Boston</td>
<td>8</td>
<td>13</td>
<td>10</td>
</tr>
<tr>
<td>Ithaca-Boston</td>
<td>22</td>
<td>20</td>
<td>18</td>
</tr>
</tbody>
</table>

The goal is to decide how many tickets from each of the 9 origin/destination/fare-class combinations to sell. The constraints are that the place cannot be overbooked on either the two legs of the flight and that the number of tickets made available cannot exceed the forecasted maximum demand. The objective is to maximize the revenue. Formulate this problem as a linear programming problem.

4. **[5 Points]** Suppose that $Y$ is a random variable taking on one of the $n$ known values:

$$a_1, a_2, \ldots, a_n.$$ 

Suppose we know that $Y$ either has distribution $p$ given by

$$P(Y = a_j) = p_j$$

or it has distribution $q$ given by

$$P(Y = a_j) = q_j.$$ 

Of course, the numbers $p_j, j = 1, 2, \ldots, n$ are nonnegative and sum to one. The same is true for the $q_j$’s. Based on a single observation of $Y$, we wish to guess whether it has distribution $p$ or distribution $q$. That is, for each possible outcome $a_j$, we will assert with probability $x_j$ that the distribution is $p$ and with probability $1 - x_j$ that the distribution is $q$. We wish to determine the probabilities $x_j, j = 1, 2, \ldots, n$, such that the probability of saying the distribution is $p$ when in fact it is $q$ has probability no larger than $\beta$, where $\beta$ is some small positive value (such as 0.05). Furthermore, given this constraint, we wish to maximize the probability that we say the distribution is $p$ when in fact it is $p$. Formulate this maximization problem as a linear programming problem.

5. **[10 Points]** Linear Programming for Graph

(a) **[2 Points]** Given a weighted, directed graph $G = (V, E)$, with weight function $w : E \to \mathbb{R}$ mapping edges to real-valued weights, a source vertex $s$, and a destination vertex $t$. Show how to compute the value $d[t]$, which is the weight of a shortest path from $s$ to $t$, by linear programming.

(b) **[3 Points]** (Based on CLRS 29.2-3)

Given a graph $G$ as in (a), write a linear program to compute $d[v]$, which is the shortest-path weight from $s$ to $v$, for each vertex $v \in V$. 

3
(c) [2 Points] Given a directed graph $G = (V, E)$, in which each edge $(u, v) \in E$ has a nonnegative capacity $c(u, v) \geq 0$. Write a linear program to compute the maximum flow $f : V \times V \to \mathbb{R}$.

(d) [3 Points] (Based on CLRS 29.2-7)
In the minimum-cost multicommodity-flow problem, we are given a 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 $\alpha(u, v)$. As in the multicommodity-flow problem (Chapter 29.2, CLRS), we are given $k$ different commodities, $K_1, K_2, \ldots, K_k$, where commodity $i$ is specified by the triple $K_i = (s_i, t_i, d_i)$. Here $s_i$ is the source of commodity $i$, $t_i$ is the sink of commodity $i$, and $d_i$ is the demand, which is the desired flow value for commodity $i$ from $s_i$ to $t_i$. We define a flow for commodity $i$, denoted by $f_i$, (so that $f_i(u, v)$ is the flow of commodity $i$ from vertex $u$ to vertex $v$) to be a real-valued function that satisfies the flow-conservation, skew-symmetry, and capacity constraints. We now define $f(u, v)$, the aggregate flow, to be sum of the various commodity flows, so that $f(u, v) = \sum_{i=1}^{k} f_i(u, v)$. The aggregate flow on edge $(u, v)$ must be no more than the capacity of edge $(u, v)$.

The cost of a flow is $\sum_{u,v \in V} f(u, v)$, and the goal is to find the feasible flow of minimum cost. Express this problem as a linear program.

6. [10 Points] Show the following problem in NP-hard.

**Problem:** INTEGER LINEAR PROGRAMMING

**Instance:** A linear program in standard form, in which $A$ and $B$ contain only integers.

**Question:** Is there a solution for the linear program, in which the $x$ must take integer values?

7. [10 Points] (Based on CLRS 30.1-7)
Consider two sets $A$ and $B$, each having $n$ integers in the range from 0 to $10n$. We wish to compute the Cartesian sum of $A$ and $B$, defined by

$$C = \{x + y : x \in A \text{ and } y \in B\}.$$

Note that the integers in $C$ are in the range from 0 to $20n$. We want to find the elements of $C$ and the number of times each element of $C$ is realized as a sum of elements in $A$ and $B$. Show that the problem can be solved in $O(n \log n)$ time. (Hint: Represent $A$ and $B$ as polynomials of degree at most 10n.)

8. [15 Points] DIVIDE-AND-CONQUER MULTIPLICATION
(Based on CLRS 30-1)

(a) [5 Points] Show how to multiply two linear polynomials $ax + b$ and $cx + d$ using only three multiplications. (Hint: One of the multiplications is $(a + b) \cdot (c + d)$.)
(b) [5 Points] Give two divide-and-conquer algorithms for multiplying two polynomials of degree-bound \( n \) that run in time \( \Theta(n^{\log 3}) \). The first algorithm should divide the input polynomial coefficients into a high half and a low half, and the second algorithm should divide them according to whether their index is odd or even.

(c) [5 Points] Show that two \( n \)-bit integers can be multiplied in \( O(n^{\log 3}) \) steps, where each step operates on at most a constant number of 1-bit values.

9. [10 Points] Given two sequences, \( a_1, \ldots, a_n \) and \( b_1, \ldots, b_m \) of real numbers, we want to determine whether there is an \( i \geq 0 \), such that \( b_1 = a_{i+1}, b_2 = a_{i+2}, \ldots, b_m = a_{i+m} \). Show how to solve this problem in \( O(n \log n) \) time with high probability.