This homework has a total of 3 problems on 3 pages. Solutions should be submitted to T-square before 12:05pm on Friday Nov 4.

The problem set is marked out of 15, you can earn up to $16 = 1 + 5 + 5 + 5$ points.

If you choose not to submit a typed write-up, please write neat and legibly. If the electronic version appears unclear, please also submit the originals, stapled, in addition to the T-square submission.

Collaboration is allowed/encouraged on problems, however each student must independently complete their own write-up, and list all collaborators. No credit will be given to solutions obtained verbatim from the Internet or other sources.

0. [1 point, only if all parts are completed]
   
   (a) Submit either a scan or photo of your homework in a single pdf file on T-square whose file name starts with your last name.
   
   (b) The answer to each question except question 0 should start on a new page.
   
   (c) Please put your name and GT-ID on each of the answer sheets.

1. [5 points] Fruitloaf Factory

   Food Inc has been told by the FDA that they must list more information on the nutrition label, and that customers will refuse to buy the product if it has too much fat or sugar. The FruitLoaf is made entirely of $n$ ingredients: $i_1, i_2, \ldots, i_n$. The original FruitLoaf has 1 gram of each ingredient $i_j$. The new FruitLoaf should be exactly $w$ grams, consisting of $g_j$ grams of ingredient $j$. Customers will not buy a FruitLoaf unless the total amount of sugar is less or equal to $S$ and the total fat is less or equal to $F$. Each ingredient $i_j$ has $f_j$ grams of fat per gram of ingredient, and $s_j$ grams of sugar per gram of ingredient.

   Customers like the old FruitLoaf flavor (without knowing what it contains), so you’re hoping to stay as close to the old recipe as possible. Customers’ taste of the difference is the Manhattan distance between the two recipes. In other words, the distance is:

   $$\sum_{j=1}^{n} |1 - g_j|$$

   (a) [1 point] Describe why a linear program cannot have

   $$\sum_{j=1}^{n} |1 - g_j|$$
as the objective.

(b) [2 points] Given the values of $w$, $S$, and $F$ as input, write a linear program to minimize the difference between the old FruitLoaf flavor profile and the new one subject to the constraints on the weight, fat content and sugar content.

(c) [2 points] The standard form of a linear program on variables $\{x_1, x_2, \ldots, x_n\}$ is in the form of:

$$\begin{align*}
\text{max} & \quad c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \\
\text{subject to} & \quad a_{(1,1)} x_1 + a_{(1,2)} x_2 + \ldots + a_{(1,n)} x_n \leq b_1 \\
& \quad a_{(2,1)} x_1 + a_{(2,2)} x_2 + \ldots + a_{(2,n)} x_n \leq b_2 \\
& \quad \vdots \\
& \quad a_{(m,1)} x_1 + a_{(m,2)} x_2 + \ldots + a_{(m,n)} x_n \leq b_m \\
& \quad x_j \geq 0 \text{ for all } 1 \leq j \leq n
\end{align*}$$

Convert your linear program from part (b) to this form.


Suppose the current football standings and the number of games remaining to be played between pairs of teams are as follows:

<table>
<thead>
<tr>
<th>Wins to Play</th>
<th>GT</th>
<th>Duke</th>
<th>Clemson</th>
<th>UGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>GT</td>
<td>13</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>Duke</td>
<td>9</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Clemson</td>
<td>8</td>
<td>7</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>UGA</td>
<td>6</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

A team finishes first if it has, or ties, for the most number of wins. Team $i$ is eliminated if there does not exist a possible outcome where it finishes first.

For example, we know for sure that UGA, with 6 wins and only 5 more games left in the season, can only hope for finishing off the season with $6 + 5 = 11$ games in the best scenario possible (if they get lucky), whereas GT already has 13 wins and can still finish off the season with better records than UGA, even under worst scenario possible of losing rest of the games scheduled. Therefore, UGA would be eliminated from contention.

Sport analysts want to find out if a certain team would be eliminated from the league. Turns out we can apply the max-flow min-cut theorem to answer this question.
(a) [1 point] Suppose we are interested in seeing if Clemson is eliminated. For each team (other than Clemson), what is the minimum number of wins they need in order to lead Clemson to elimination? (Hint: think of the maximum possible number of wins that Clemson can have by the end of the season)

(b) [2 points] Consider the following ‘greedy’ like algorithm for finding the eliminated teams:

i. For each team $i$, compute $W_{i}^{\text{max}}$ and $W_{i}^{\text{min}}$, the maximum and minimum number of games that team $i$ could win assuming that it wins/looses all of its remaining games respectively.

ii. Declare a team $i$ to be eliminated if and only if $w_{i}^{\text{max}} < w_{j}^{\text{min}}$ for some $j \neq i$.

In other words, this algorithm declares a team $i$ to be not eliminated if for every other team $j$, $i$ can end up ahead of $j$ if $i$ wins all of its remaining games and $j$ looses all of its remaining games.

Give an example with at most 4 teams where a team that this algorithm declares as not-eliminated is actually eliminated.

(c) [2 points] Explain how to use maximum flow to determine if a team is eliminated. Bound the size of the flow network when $n$ teams are involved.

3. [5 points] Return of the Joker

The Joker is considering which heists to perform tomorrow from a set $H = \{H_1, H_2, \ldots, H_m\}$ with payoffs $P = \{p_1, p_2, \ldots, p_m\}$. He has access to a set of minions $A = \{A_1, A_2, \ldots, A_n\}$ who charge daily fees of $C = \{c_1, c_2, \ldots, c_n\}$. Each heist $H_j$ requires some subset of minions $R_j \subseteq A$. One minion can work in multiple heists, and it is possible to perform all heists in $H$ in one day.

The joker’s task is to find an algorithm to determine which heists to perform and which minions to hire to maximize his profit, which is the total payoffs of the heists performed minus the cost of hiring the minions.

Consider the following network $G$:

- The network contains source vertex $s$, vertices $A_1, A_2, \ldots, A_n$, vertices $H_1, H_2, \ldots, H_m$, and a sink vertex $t$.
- For $k = 1, 2, \ldots n$, there is an edge $(s, A_k)$ of capacity $c_k$, and
- for $j = 1, 2, \ldots m$, there is an edge $(H_j, t)$ of capacity $p_j$.
- For $k = 1, 2, \ldots n$ and $j = 1, 2, \ldots m$, if $A_k \in R_j$, then there is an edge $(A_k, H_j)$ of infinite capacity.

We want to show that the optimum set of heists to perform can be deduced from the minimum cut on this network.
(a) [1 point] Consider some cut \((S, T)\) of \(G\) with finite capacity. Show that if \(H_j \in T\), then \(A_k \in T\) for each \(A_k \in R_j\).

(b) [2 points] Show how to determine the maximum profit from the minimum \(s \rightarrow t\) cut of \(G\), and the values \(c_i\) and \(p_j\).

(c) [2 points] Let \(p_{\text{total}} = \sum_{j=1}^m p_j\). Give an \(O(n^{20} m^{20} p_{\text{total}}^{20})\) time algorithm to determine the optimum set of heists to perform and minions to hire. Analyze the computational complexity of your algorithm in terms of \(m, n\), and \(p_{\text{total}}\).