MA 351 Fall 2024 (Aaron N. K. Yip)
Homework 3
Due: Thursday, Sept. 12, in class

Penney, Linear Algebra: Ideas and Applications (4th edition)
Section 1.4 EXERCISES, p. 86: 1.93, 1.96, 1.108, 1.109, 1.110, 1.111, 1.112, 1.121

Hint/approach for 1.108, 1.109, 1.111, 1.112. To show that $\text{Span}\{u_1, u_2, \ldots, u_k\} = \text{Span}\{v_1, v_2, \ldots, v_l\}$ – note that in general $k \neq l$, you need to show the following two steps:

1. for *every* $s_1, s_2, \ldots s_k$, you can find $t_1, t_2, \ldots t_l$ such that

$$t_1 v_1 + t_2 v_2 + \cdots t_l v_l = s_1 u_1 + s_2 u_2 + \cdots s_k u_k,$$

and

2. for *every* $t_1, t_2, \ldots t_l$, you can find $s_1, s_2, \ldots s_k$ such that

$$s_1 u_1 + s_2 u_2 + \cdots s_k u_k = t_1 v_1 + t_2 v_2 + \cdots t_l v_l.$$

Essentially, you can relate every $(s_1, s_2, \ldots s_k)$ to $(t_1, t_2, \ldots t_l)$ and vice versa. Alternatively, you can also show that

1. *every single one of* u_i can be written as linear combination of the v_j's,

and

2. *every single one of* v_j can be written as linear combination of the u_i's.

Hint/approach for 1.110. This is similar to the above. Note that the first set of s, t *might be different* from the second set of s, t. Essentially, you need to show that

1. For *every* s_1, t_1, you can find s_2, t_2 such that

$$\begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} + s_2 \begin{pmatrix} -4 \\ 1 \\ 2 \end{pmatrix} + t_2 \begin{pmatrix} -3 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + s_1 \begin{pmatrix} -3 \\ 1 \\ 1 \end{pmatrix} + t_1 \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

and

2. For *every* s_2, t_2, you can find s_1, t_1 such that

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + s_1 \begin{pmatrix} -3 \\ 1 \\ 1 \end{pmatrix} + t_1 \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} + s_2 \begin{pmatrix} -4 \\ 1 \\ 2 \end{pmatrix} + t_2 \begin{pmatrix} -3 \\ 1 \\ 1 \end{pmatrix}.$$

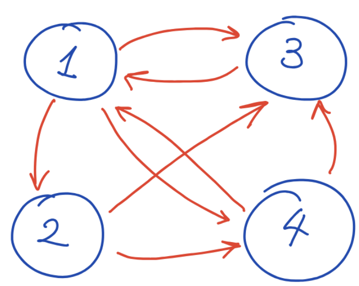
Additional Problem. This problem lets you have a peek into the inner workings of the Google PageRank algorithm, in its most primitive form, uncorrupted by any human interference. It also shows that *homogeneous* equation $AX = 0$ is *not unimportant.*

The output of the PageRank algorithm is the ranking or *relative* importance of interconnected webpages. The (relative) importance of a webpage is given by a positive number and is determined by how many *other* webpages point to it. Let the (whole) internet consists of N webpages (nodes). Let x_i denote the (relative) importance of node i. Then for $i = 1, 2, \dots N$, x_i is given by:

$$x_i = \sum_j \frac{x_j}{D_j},$$

where the summation is over all the *other* webpages j that has a link to i and D_j is the total number of links (*degree*) that point out of j. The division by D_j is such that if j points out to (million) many other webpages, then its contribution to the importance of any webpage it points to will be diminished. (We will assume that we do not consider any self-referencing of the webpages.)

Consider the following example of an internet with four webpages (nodes):



For example, Webpage 1 points out to Webpages 2, 3, and 4 so that $D_1 = 3$, and only Webpages 3 and 4 point into Webpage 1. For Webpage 2, only Webpage 1 points to it and it points out only to Webpages 3 and 4 so that $D_2 = 2$. Note also that $D_3 = 1$ and $D_4 = 2$. Now let x_1, x_2, x_3, x_4 be the (relative) importance of the four webpages. Then we have

$$\begin{aligned} x_1 &= \qquad\quad x_3 + \tfrac{x_4}{2} \quad ; \\ x_2 &= \tfrac{x_1}{3} \qquad\qquad\qquad ; \end{aligned}$$

Write down the corresponding equations for x_3 and x_4. Put the overall 4×4 system into a homogenous form $AX = 0$. Solve for x_1, x_2, x_3, x_4. Note that you should be able to express your solution in terms of *one* free variable t. Normalize your solution by choosing t such that $x_1 + x_2 + x_3 + x_4 = 1$. Find the corresponding x_1, x_2, x_3, x_4 using this t. Then the ranking of the four webpages is given by the ranking of the x_i's, i.e. the larger is the x_i, the more important is Webpage i.

(You should be able to do this problem just with the description given above. But if you are curious, you can look at the article, *The* $25,000,000,000 *eigenvector: The linear algebra behind Google*, by K. Bryan and T. Leise.)