1. **Solve It**

   Solve the following system of linear equations

$$\begin{bmatrix} 4 & -6 \\ -8 & 9 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 18 \\ -32 \\ 1 \end{bmatrix}$$

2. **Invert It**

   What is the inverse of the matrix $\begin{bmatrix} 2 & -4 \\ 1 & 6 \end{bmatrix}$?

3. **Null It**

   What is the null space of the matrix $\begin{bmatrix} 4 & 2 \\ -5 & 7 \end{bmatrix}$?

4. **Null It Again**

   What is the null space of the matrix $\begin{bmatrix} 3 & 15 \\ -5 & -25 \end{bmatrix}$?

5. **Null It Yet Again**

   What is the null space of the matrix $\begin{bmatrix} 1 & 1 & 0 & 3 & 0 \\ 0 & 0 & 1 & 4 & -2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$?

6. **Permute It**

   (a) Given a vector $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$, find a matrix that permutes the components of $\vec{x}$ to $\vec{x'} = \begin{bmatrix} x_3 \\ x_5 \\ x_1 \\ x_2 \\ x_4 \end{bmatrix}$. That is, **find a matrix A such that $\vec{x'} = A\vec{x}$**

   (b) **Is the matrix A invertible? If yes, find its inverse, A^{-1}, so that $A^{-1}\vec{x'} = \vec{x}$.**

7. **Show it**

   Let n be a positive integer. Given a vector $\vec{v} \in \mathbb{R}^n$, consider the set $V^\perp = \{\vec{w} \in \mathbb{R}^n \mid \langle \vec{w}, \vec{v} \rangle = 0\}$. This means that $V^\perp$ is the set of all vectors $\vec{w} \in \mathbb{R}^n$ such that $\langle \vec{w}, \vec{v} \rangle = 0$. Show that $V^\perp$ is a subspace of $\mathbb{R}^n$.

8. **Show It Again**

   Let $\vec{a}_1^T$ be the first row of matrix A. Let $\vec{x}$ be a vector in the nullspace of A. **Show that $\langle \vec{a}_1, \vec{x} \rangle = 0$.**

9. Sparse Imaging (17pts)

You want to take a 3×3 image of an unknown sample, depicted below. Assume the 9 pixels of the sample can take on any real value, so the sample can be represented by a vector $\vec{x} \in \mathbb{R}^9$.

x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

Like the lab, you need to make multiple exposures to recover the whole image. In each exposure, you can choose to expose any set of pixels, and you will measure the sum of the chosen pixel values. For example, you can expose pixels $1,2$ and 5 simultaneously, and will measure the value $(x_1 + x_2 + x_5)$.

In your first attempt, you image the sample by simply exposing each pixel individually, for a total of 9 exposures. This works, but occasionally runs into a problem: Sometimes one exposure fails (say, the film gets corrupted). You know when an exposure fails, but there is a large turn-around time to get film developed, so you want to avoid going back and re-imaging the failed pixel again. Instead, you would like to take an additional exposure (for 10 total, instead of 9), such that even if one of them fails, you can still recover the image.

(a) **(7pts)** For your additional exposure, you decide to expose all the pixels. That is, you measure $x_1 + x_2 + \cdots + x_9$. **Show that you can always recover all 9 pixels of the image, even if one exposure (out of the 10 total exposures) fails.**

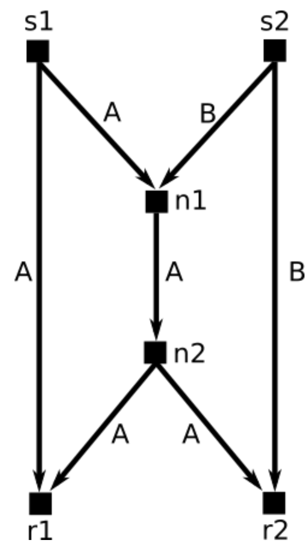
(b) **(10pts)** Illuminating each pixel requires a light source, so you want to minimize the number of pixels exposed in your additional exposure. For example, would it suffice to expose only half of the pixels, instead of all the pixels as in part (a)? **Show that in fact, the additional exposure *must* expose all pixels in order to tolerate any one of your exposures failing.**

10. Network Coding (27 + 10pts)

[**NOTE: The last part of this problem is extra credit. Do it if you have time, but don't get stuck on it if you don't.**]

Suppose we have the following network consisting of routers and wires connecting the routers. In this problem, we assume that a message is just a real number. Each of the senders wants to broadcast their own message and each wants to make sure that **all** the receivers get their message. Each of the receivers wants to be able to decode all of the messages.

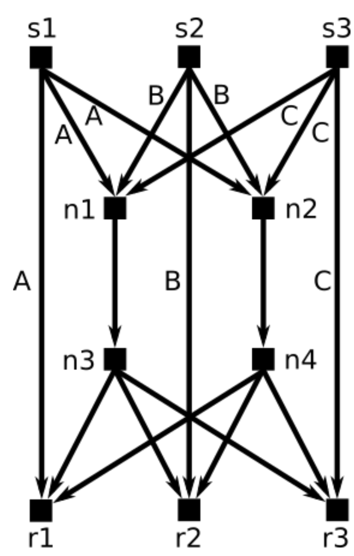
Each wire can carry a message in the direction indicated, and each router can receive many messages but can only broadcast a single message to the wires coming out of it. Suppose each router only gets to broadcast once. For example, senders s_1 and s_2 send out messages A and B respectively. Router n_1 receives both A and B, but in this example chooses to broadcast A. n_2 re-broadcasts whatever it gets from n_1.



(a) **(4pts)** In the routing scheme described above, **can both receivers r_1 and r_2 recover both messages A and B? If yes, describe how. If no, explain why not.**

(b) **(8pts)** Alice comes up with a brilliant idea. If n_1 sends the single real number $A+B$ instead of A, **can we now recover both A and B at both receiver nodes? If yes, describe how. If no, explain why not.**

(c) **(15pts)** Now we have a network with 3 senders and 3 receivers. s_1 sends message A, s_2 sends message B and s_3 sends message C. The two intermediate routers n_1 and n_2 each receive A, B and C. They can choose to send any message. The routers n_3 and n_4 can only re-broadcast the single message they receive.



Suppose that the routers n_1 and n_2 broadcast a linear combination of the messages they receive. If n_1 broadcasts $2A-B+4C$ and n_2 broadcasts $A+3B+2C$, **can each receiver recover all 3 messages? If yes, describe how. If no, explain why not.**

(d) **(BONUS 10pts)** Suppose now that n_1 broadcasts $2A-3B+C$ and n_2 broadcasts $A+3B+2C$, **can each receiver recover all 3 messages? If yes, describe how. If no, explain why not.**

11. **How Stuff (Netflix) Works (32pts)**

Most of you've used on-demand Internet movie providers, like Netflix. You might have noticed that Netflix gives you a list of recommended movies, presumably based on the movies you have already reviewed.

But how does Netflix determine which movies you might want to watch? Even when you watch movies, you're often too lazy to rate them, and Netflix therefore might actually only have a handful of ratings it can use to decide which movies to recommend to you. **Out of the tens of thousands of movies on Netflix, how do they figure out which you like and dislike from so little information?**

Let's try to answer this question with an overly simplified toy model:

We denote the total number of movies on Netflix by n (in the order of tens of thousands), and your "score" for movie i by ℓ_i. For simplicity in the model, let's assume that ℓ_i can be any real number. (The higher the value of ℓ_i, the more you like the movie. Negative numbers represent dislikes, and the number 0 represents that you are neutral.) Then, we can represent your overall movie preferences by the "score" vector:

$$\vec{\ell} = \begin{bmatrix} \ell_1 \\ \ell_2 \\ \vdots \\ \ell_n \end{bmatrix} \tag{1}$$

You might think that Netflix needs to know the scores you associate with all n movies to make recommendations. But here's the catch: Netflix can assume that your "score" vector has specific structure. One instance of such structure is that your scores for two movies in the Lord of the Rings saga are likely to be the same. (Or, if you hate a movie with Nicholas Cage in it, you're likely to hate *all* his movies.)

Mathematically, we assume that a valid score vector $\vec{\ell}$ will always lie in a much smaller k-dimensional *subspace* of $\mathbb{R}^n$. (k could be as small as in the order of tens!) Let's assume Netflix knows some basis vectors $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k \in \mathbb{R}^n$ that *span* this subspace. Recall, basis vectors are *linearly independent*. That is, we can express any $\vec{\ell}$ as a unique linear combination of the basis vectors as follows:

$$\vec{\ell} = \sum_{i=1}^{k} \alpha_i \vec{u}_i \tag{2}$$

(You could think of the basis vectors capturing the preferences of users with very ideologically pure likes or dislikes. They could correspond to users who like only one genre of movies, fans of a particular actor, etc.)

(a) **(5pts)** For a given score vector $\vec{\ell}$, we would like to determine the coefficients $\alpha_1, \alpha_2, \ldots, \alpha_k$. Write a system of equations in matrix form, relating some matrix U, $\vec{\ell}$ and $\vec{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix}$. What is the dimension of the column space of U?

(b) **(12 pts)** Outline a procedure by which we can determine a set of k movies whose scores should be measured to determine $\vec{\alpha}$. That is, determine which components of $\vec{\ell}$ are sufficient to measure in order to fully determine $\vec{\alpha}$. Express your answer in terms of the matrix U.

(Hint: Gaussian Elimination might come in handy.)

(c) **(10pts)** Given a set of k movies (chosen according to the procedure you gave in the earlier part) whose scores are measured, how would you solve for $\vec{\alpha}$ if you know only these k components of $\vec{\ell}$ (k movie scores)?

(d) **(5pts)** Use the answer you obtained in part (c) to obtain the entire score vector, $\vec{\ell}$. Congratulations! You have magically determined the entire preference vector ($\vec{\ell}$) from a handful of scores ($\vec{\ell}^{(k)}$).