EECS 16A   Designing Information Devices and Systems I
Fall 2020   Midterm 1 Instructions

Goodluck for the midterm! You've studied hard and we are rooting for you to do well! Please read these instructions and the proctoring guidelines before the exam.

Our advice to you: if you can't solve a particular problem, move on to another, or state and solve a simpler one that captures at least some of its essence. You will perhaps find yourself on a path to the solution. **Good luck! We believe in you.**

Format & How to Submit Answers

There are 7 problems (2 introductory questions, and 5 exam questions with subparts) of varying numbers of points on the exam. The problems are of varying difficulty, so pace yourself accordingly and avoid spending too much time on any one question until you have gotten all of the other points you can. If you are having trouble with one problem, there may be easier points available later in the exam!

Complete your exam using either the template provided or appropriately created sheets of paper. Either way, you should submit your answers to the Gradescope assignment that is marked Midterm 1 for your specific exam group. Make sure you submit your assignment to the correct Gradescope assignment. You MUST select pages for each question. We cannot grade your exam if you do not select pages for each question. If you are having technical difficulties submitting your exam, you can email your answers to eecs16a@berkeley.edu.

In general, show all your work legibly to receive full credit, we cannot grade anything that we cannot read. For some problems, we may try to award partial credit for substantial progress on a problem; and showing your work clearly and legibly will help us do that.

Timing & Academic Honesty

You are expected to follow the rules provided in the Exam Proctoring Guidelines (`https://docs.google.com/document/d/1EVb4Ca6FWSAykExY7X5ynFW4KdmHd0BI6KZ0ktM8ows/edit?usp=sharing`). The exam will be available to you at the link sent to you by email. The exam will start at 7pm Pacific Time, Monday, Oct 5th 2020, unless you have an exam accommodation. If you experience technical difficulties and cannot access your exam, let us know by making a private post on Piazza and we will try to help.

You have a 120 minutes for the exam, with extra time for scanning and submitting to Gradescope (20 minutes for a paper answer sheet, 10 minutes for a tablet.) Most of you will have to submit your exam by 9:10pm (if you are using a tablet) or 9:20pm (if you are using paper), unless you have another accommodation. Late submissions will be penalized exponentially. An exam that is submitted N minutes after the end of the submission period will lose 2^N points. This means that if you are 1 minute late you will lose 2 points; if you are 5 minutes late you will lose 32 points and so on.

This is a closed-note, closed-book, closed-internet, and closed-collaboration exam. Calculators are not allowed. You may consult a single handwritten 8.5" by 11" cheat sheet (front and back). Do not attempt to cheat in any way. We have a zero tolerance policy for violations of the Berkeley Honor Code.

EECS 16A   Designing Information Devices and Systems I
Fall 2020   Midterm 1

1. **HONOR CODE**

   If you have not already done so, please copy the following statements into the box provided for the honor code on your answer sheet, and sign your name.

   I will respect my classmates and the integrity of this exam by following this honor code. I affirm:

   - *I have read the instructions for this exam. I understand them and will follow them.*
   - *All of the work submitted here is my original work.*
   - *I did not reference any sources other than my one reference cheat sheet.*
   - *I did not collaborate with any other human being on this exam.*

2. (a) **Tell us about something you did in the last year that you are proud of.** *All answers will be awarded full credit; you can be brief.* **(2 Points)**

   (b) **Name someone that inspires you.** *All answers will be awarded full credit; you can be brief.* **(2 Points)**

3. Sailing for climate change (22 points)

You are a member of Greta Thunberg's crew that embarks on a new transatlantic boat journey, advocating for climate change awareness and environmentally friendly travel.

(a) (2 points) In the environmentally friendly mode, the ship travels using only the wind and sail; as a result you have to move in direction of the wind.

In this mode, the next location of the ship is given by $\vec{\ell}_{next} = \mathbf{A} \cdot \vec{\ell}_{curr}$, where $\vec{\ell}_{curr} = \begin{bmatrix} 2 \\ 5 \\ 0 \end{bmatrix}$ is the current location of the ship. $\mathbf{A}$ is given as:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 0 \\ 3 & 6 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Find your next location vector $\vec{\ell}_{next}$ after you apply the steering matrix A once to $\vec{\ell}_{curr}$.

(b) (2 points) **What is the dimension of the columnspace of A? Explain your answer and/or show your work.**

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 0 \\ 3 & 6 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

(c) (6 points) A plane is monitoring the journey from a high altitude and wants to come closer to Greta's ship at a certain meeting point, $\vec{\ell}_{meet}$.

The plane's current location is described by $\vec{\ell}_{plane} = \begin{bmatrix} 0 \\ 20 \\ 100 \end{bmatrix}$, and the goal is to hover at the meeting point described by $\vec{\ell}_{meet} = \begin{bmatrix} 20 \\ 20 \\ 50 \end{bmatrix}$. You have been asked to design a steering matrix $\mathbf{B}$ for the plane so that

$$\vec{\ell}_{meet} = \mathbf{B} \cdot \vec{\ell}_{plane}.$$

Some of the entries of the matrix $\mathbf{B}$ are given for you below; however, **you must find the elements p, q and r.** Please draw a box around your answers for p, q and r.

$$\mathbf{B} = \begin{bmatrix} 10 & p & 1 \\ 20 & 6 & q \\ 7 & r & -1 \end{bmatrix}.$$

(d) (4 points) **For this subpart and the following subparts, you can assume that your ship travels only in 2D.**

Your current location is given by $\vec{\ell}_{curr} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$. Steering matrix $\mathbf{C}$ was multiplied once with your previous location vector, $\vec{\ell}_{prev}$, to bring you to $\vec{\ell}_{curr}$, so you know $\vec{\ell}_{curr} = \mathbf{C} \cdot \vec{\ell}_{prev}$. The matrix $\mathbf{C}$ is:

$$\mathbf{C} = \begin{bmatrix} 1 & 1 \\ 3 & 6 \end{bmatrix}.$$

Find $\vec{\ell}_{prev}$, your previous location vector.

(e) (4 points) On the distant horizon, you see three other ships approaching Greta's ship. The current locations of the three ships are given by the column vectors: $\begin{bmatrix} 5 \\ 9 \end{bmatrix}$, $\begin{bmatrix} 4 \\ 12 \end{bmatrix}$, $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$. You stack the ships' locations together to make the matrix $\mathbf{D} = \begin{bmatrix} 5 & 4 & 3 \\ 9 & 12 & 4 \end{bmatrix}$.

The winds driving these ships transform each ship location by the matrix $\mathbf{E} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$.

Instead of computing their final locations one-by-one, you want one matrix that gives the locations of the three ships, where the two rows of this matrix will contain the x- and y-coordinates of the ships, respectively.

Set this problem up as a matrix-matrix multiplication and find the matrix representing the final locations of all three ships.

(f) (4 points) Your current location vector is now given as $\vec{\ell}_{curr} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$. Your would like to steer Greta's ship to $\vec{\ell}_{final} = \begin{bmatrix} 0 \\ -10 \end{bmatrix}$. You notice that the ship has several built-in 2×2 steering matrices which you can apply in any order to your current location vector:

- $\mathbf{T}_\theta$, which rotates your current location vector by $90°$ counterclockwise;
- $\mathbf{T_2}$, which multiplies your current location vector by a factor of 2;
- $\mathbf{T_x}$, which reflects your current location vector across the x-axis;
- $\mathbf{T_y}$, which reflects your current location vector across the y-axis.

You would like to only use these built-in steering matrices to direct Greta's ship to its final goal.

Which of the following options would successfully get you from $\vec{\ell}_{curr}$ to $\vec{\ell}_{final}$? Only one option is correct.

No partial credit will be awarded if any incorrect options are selected. You do not need to show any work for this problem.

(i) $\vec{\ell}_{final} = \mathbf{T}_\theta \mathbf{T_2} \mathbf{T_x} \vec{\ell}_{curr}$ 　　　　　　(iii) $\vec{\ell}_{final} = \mathbf{T_y} \mathbf{T_2} \mathbf{T_x} \vec{\ell}_{curr}$

(ii) $\vec{\ell}_{final} = \mathbf{T_x} \mathbf{T_2} \mathbf{T}_\theta \vec{\ell}_{curr}$ 　　　　　　(iv) $\vec{\ell}_{final} = \mathbf{T_y} \mathbf{T_x} \mathbf{T_2} \vec{\ell}_{curr}$

4. Mixing Paints (21 points)

Over quarantine, you've really gotten into painting, but you are running low on paints. You would like to mix the paints you have to make different colors.

Notation: **Every** color can be represented as a length-3 vector, $\vec{c} = \begin{bmatrix} c_r \\ c_y \\ c_b \end{bmatrix}$, where c_r, c_y, and c_b represent the number of bottles of red, yellow, and blue paint, respectively, to make the color.

(a) (5 points) You want to make the color brown, $\vec{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, using a linear combination of turquoise, magenta, and peach paints (the only ones you have left). We are given the following information about paint compositions:

- 1 bottle of turquoise is made by combining 0 bottles red, 0.4 bottles yellow, and 0.6 bottles blue.
- 1 bottle of magenta is made by combining 0.5 bottles red, 0 bottles yellow, and 0.5 bottles blue.
- 1 bottle of peach is made by combining 0.5 bottles red, 0.4 bottles yellow, and 0.1 bottles blue.

You would like to find the number of bottles of turquoise (x_t), magenta (x_m), and peach (x_p) to mix to make brown. Formulate this problem as a matrix-vector equation. You do not need to solve the equation.

(b) (5 points) Your friend wants to make the color brown but has a different set of colors. She also sets up a system of equations to make brown and ends up with $\mathbf{A}\vec{x} = \vec{b}$ where

$$\mathbf{A} = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0 \end{bmatrix}, \vec{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Use Gaussian Elimination to solve for $\vec{x}$, which represents the mixture of paints your friend must combine to make brown. Show your work.

(c) (6 points) You now want to make lots of colors. Instead of solving the system $\mathbf{A}\vec{x} = \vec{c}$ every time we want to make a new color $\vec{c}$, you want to find the inverse matrix $\mathbf{A}^{-1}$. For this subpart, use the following matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix}.$$

Use Gaussian Elimination to determine if $\mathbf{A}^{-1}$ exists. If $\mathbf{A}^{-1}$ exists, give its value.

(d) (5 points) We would like to determine what colors we can make just given three other colors (purple, orange, gray). These are represented by a matrix $\mathbf{D} \in \mathbb{R}^{3 \times 3}$, which is invertible, but is otherwise unknown to you. You cannot mix together negative amounts of paint.

Consider the set of all colors

$$\mathbb{S} = \left\{ \vec{c} \in \mathbb{R}^3 \mid \quad \mathbf{D}\vec{x} = \vec{c}, \ \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \right\}.$$

that we can get from mixing together non-negative quantities of the three colors. **Is $\mathbb{S}$ a vector subspace of $\mathbb{R}^3$? Justify your answer.**

5. Basis and Linear Independence (18 points)

Let $\mathbb{V}$ be a subspace of $\mathbb{R}^4$ defined as

$$\mathbb{V} = \left\{ \vec{x} \in \mathbb{R}^4 \quad \middle| \quad \vec{x} = \begin{bmatrix} \alpha \\ \beta \\ \alpha + \beta \\ \alpha - \beta \end{bmatrix}, \text{ where } \alpha, \beta \in \mathbb{R} \right\}.$$

(a) (4 points) **Find a basis for $\mathbb{V}$ and determine the dimension of $\mathbb{V}$. Show your work.**

(b) (4 points) Given $\vec{v}_1 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$ and $\vec{v}_2 = \begin{bmatrix} 2 \\ 5 \\ 3 \end{bmatrix}$, **determine if the set $\{\vec{v}_1, \vec{v}_2\}$ is linearly independent, using Gaussian Elimination.**

(c) (10 points) **Prove the following statement**: If

$$\{\vec{v}_1, \vec{v}_2, \vec{v}_3, \cdots, \vec{v}_n\}$$

is a linearly independent set, then

$$\{\vec{v}_1, \vec{v}_2 - \vec{v}_1, \vec{v}_3 - \vec{v}_2 - \vec{v}_1, \cdots, \vec{v}_n - \vec{v}_{n-1} - \cdots - \vec{v}_2 - \vec{v}_1\}$$

is a linearly independent set.

6. GPT-3 wrote this question... or did it? (33 points)

Your friend, a UC Berkeley undergraduate, is working on a project to use OpenAI's GPT-3 language gener-
ation algorithm to write blog posts that mimic human writers. After one of their first posts trends on Hacker
News, you apply for access to GPT-3 so you can examine how it works for yourself.

The algorithm works by iterating. The blog post at time step n is represented by a vector $\vec{x}[n]$. We will define
the entries of the vector later. The GPT-3 algorithm is represented by the transition matrix $\mathbf{A}$. So the blog
post at time $n+1$ is given by:

$$\vec{x}[n+1] = \mathbf{A} \cdot \vec{x}[n].$$

(a) (4 points) You find that the transition matrix $\mathbf{A}$ of one version of the algorithm is described by the flow
diagram in **Figure 6.1**.
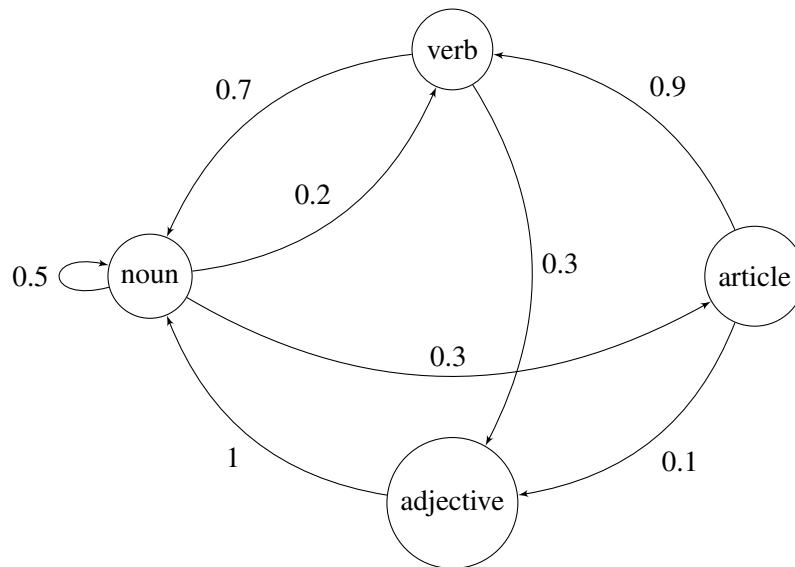
Figure 6.1: A flow diagram to represent how model $\mathbf{A}$ transforms state vector $\vec{x}[n]$.

In this case, each entry of $\vec{x}[n]$ describes the number of nouns, verbs, adjectives and articles in the blog
post at time n. The structure of the state vector $\vec{x}[n]$ for $n \geq 0$ is:

$$\vec{x}[n] = \begin{bmatrix} x_{\text{noun}}[n] \\ x_{\text{verb}}[n] \\ x_{\text{adjective}}[n] \\ x_{\text{article}}[n] \end{bmatrix}. \tag{1}$$

What is the A matrix based on the flow diagram in Figure 6.1?

(b) (4 points) Now, you are interested in testing out a less complex model that operates on just three parts of speech: nouns, verbs, and adjectives. In this case, the structure of the state vector $\vec{x}[n]$ at each iteration n is:

$$\vec{x}[n] = \begin{bmatrix} x_{\text{noun}}[n] \\ x_{\text{verb}}[n] \\ x_{\text{adjective}}[n] \end{bmatrix}.$$

Your starting article, $\vec{x}[0]$, consists of 4000 nouns, 4000 verbs, and 4000 adjectives, i.e, $\vec{x}[0] = \begin{bmatrix} 4000 \\ 4000 \\ 4000 \end{bmatrix}.$

You want to try out a different 3×3 transition matrix $\mathbf{B}$ to see what happens to your starting article.

$$\mathbf{B} = \begin{bmatrix} 0.25 & 0.25 & 0 \\ 0.25 & 0 & 0.25 \\ 0 & 0.25 & 0.25 \end{bmatrix}.$$
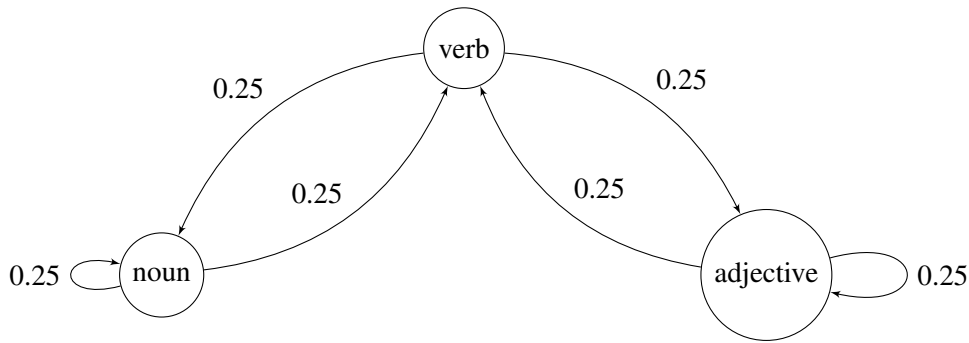
Figure 6.2: A flow diagram to represent how model $\mathbf{B}$ transforms state vector $\vec{x}[n]$.

Compute the next state vector, $\vec{x}[1]$, if we apply matrix B once to the starting state vector, $\vec{x}[0]$. Show your work.

(c) (2 points) **Is the transition matrix B representative of a conservative system? Why or why not?**

$$\mathbf{B} = \begin{bmatrix} 0.25 & 0.25 & 0 \\ 0.25 & 0 & 0.25 \\ 0 & 0.25 & 0.25 \end{bmatrix}$$

(d) (4 points) Now, you are considering testing out an unknown 3×3 model matrix $\mathbf{C}$. You have limited information about $\mathbf{C}$: you know that $\mathbf{C}$ has three eigenvectors $\vec{v}_1$, $\vec{v}_2$, and $\vec{v}_3$, and three corresponding eigenvalues λ_1, λ_2 and λ_3.

You are also told that you can rewrite $\vec{x}[0]$, the starting state vector, as a linear combination of the eigenvectors of $\mathbf{C}$:

$$\vec{x}[0] = -3000\vec{v}_1 + 3000\vec{v}_2 + 7000\vec{v}_3.$$

You are ultimately interested in observing how $\mathbf{C}$ modifies the state vector, $\vec{x}[n]$, at every timestep n.

Write an expression for $\vec{x}[n]$ in terms of n, the eigenvalues of C, and the eigenvectors of C. You do not have to show a lot of work for this problem. Your answer may also include constants. *Note: Your equation **should not** include any terms involving previous states (in other words, do not use terms such as $n-1$, $n-2$, ...).*

(e) (6 points) Suppose you are now working with a 7000-word blog post consisting entirely of nouns, i.e.,

$$\vec{x}[0] = \begin{bmatrix} x_{\text{noun}}[0] \\ x_{\text{verb}}[0] \\ x_{\text{adjective}}[0] \end{bmatrix} = \begin{bmatrix} 7000 \\ 0 \\ 0 \end{bmatrix}.$$

You'd like to add some variety to the parts of speech used in your article and introduce some verbs and adjectives, too. Someone on the internet suggests using the following matrix **D**:

$$\mathbf{D} = \begin{bmatrix} 0.3 & 0 & 0.4 \\ 0.4 & 1 & 0.2 \\ 0.3 & 0 & 0.4 \end{bmatrix}.$$

The eigenvalues of **D** are $\lambda_1 = 0$, $\lambda_2 = 0.7$, and $\lambda_3 = 1$, with corresponding eigenvectors

$$\vec{v}_1 = \begin{bmatrix} -\frac{4}{3} \\ \frac{1}{3} \\ 1 \end{bmatrix}, \vec{v}_2 = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}, \text{ and } \vec{v}_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

Recall that $\vec{x}[n+1] = \mathbf{D} \cdot \vec{x}[n]$.

What is the steady-state of the system, i.e., $\lim_{n \to \infty} \vec{x}[n]$?

Hint: You do NOT have to express $\vec{x}[0]$ in terms of $\vec{v}_1, \vec{v}_2, \vec{v}_3$ to solve this part. Also, think about if the system is conservative.

(f) (5 points) You are still considering new ways to make your starting 7000-word blog post consisting of all nouns (i.e, $\vec{x}[0] = \begin{bmatrix} 7000 \\ 0 \\ 0 \end{bmatrix}$) more interesting. Your EECS16A TA, Lily, suggests the following transition matrix **E**, and tells you its eigenvalues are $\lambda_1 = 0.4$, $\lambda_2 = 0.1$, and $\lambda_3 = 1$.

$$\mathbf{E} = \begin{bmatrix} 0.3 & 0.1 & 0.2 \\ 0.4 & 0.8 & 0.4 \\ 0.3 & 0.1 & 0.4 \end{bmatrix}$$

Compute the eigenspace corresponding to the eigenvalue $\lambda_2 = 0.1$ for matrix E. Show your work.

(g) (8 points) You are finally considering a model **F** matrix:

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}.$$

The entries of the columns of this matrix add up to 2, i.e.:

$$f_{11} + f_{21} + f_{31} = 2, \qquad f_{12} + f_{22} + f_{32} = 2, \qquad f_{13} + f_{23} + f_{33} = 2.$$

What does this transition matrix do to the **total number of words (i.e. verbs + nouns + adjectives) in each iteration** of the blog post? **Provide a rigorous proof of your answer.** Recall that $\vec{x}[n+1] = \mathbf{F} \cdot \vec{x}[n]$, and that

$$\vec{x}[n] = \begin{bmatrix} x_{\text{noun}}[n] \\ x_{\text{verb}}[n] \\ x_{\text{adjective}}[n] \end{bmatrix}.$$

7. **Get in the robot, 16A people! (31 points)**

   You are testing a giant robot that you programmed.

   (a) (8 points) You can use a navigation control input $\vec{u} \in \mathbb{R}^3$ to move the robot to different positions $\vec{p}$, as described by:

   $$\mathbf{A}\vec{u} = \vec{p}, \quad \text{where} \quad \mathbf{A} = \begin{bmatrix} 1 & 0 & 2 \\ 1 & 1 & 2 \\ 0 & 1 & a \end{bmatrix}.$$

   The value of the parameter $a \in \mathbb{R}$ determines the robot's behavior.

   You find your giant robot moving only along a plane in space defined by

   $$\mathbb{U} = \left\{ \vec{p} \in \mathbb{R}^3 \;\middle|\; \vec{p} = \begin{bmatrix} \alpha \\ \alpha + \beta \\ \beta \end{bmatrix}, \quad \alpha, \beta \in \mathbb{R} \right\}.$$

   Find the value of a in the programmed matrix, which caused your robot to get stuck on the plane $\mathbb{U}$. Show your work and justify your answer.

   (b) (6 points) Your robot operates with the same model as in part (a), i.e. $\mathbf{A}\vec{u} = \vec{p}$. We recall $\vec{u}$ is a navigation control input and $\vec{p}$ is the robot's position. However, we lost the documentation about the $\mathbf{A}$ matrix! In other words, **we don't know what A is and it may not be the same matrix as in part (a)**.

   You found an interesting phenomenon: two different navigation control inputs, $\vec{u}_1$ and $\vec{u}_2$ move the robot to the same position $\vec{p}$, i.e.

   $$\mathbf{A}\vec{u}_1 = \mathbf{A}\vec{u}_2 = \vec{p}, \quad \text{where} \quad \vec{u}_1 = \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}, \quad \vec{u}_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \text{and} \quad \vec{p} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}.$$

   With this information, **find a control input $\vec{u}$ that can return you back to $\vec{p} = \vec{0}$**, where your base of operations is located. Due to some limitations of the robot motors, **you cannot command a zero control input, i.e. $\vec{u} \neq \vec{0}$. Show your work and justify your answer.**

   (c) (5 points) Another challenge in navigation is collision avoidance — you don't want to crash.

   You are given the following documentation about the range sensor on Robot A, which is the robot you control. The range sensor can only detect other robots in the following subspace $\mathbb{S}$:

   $$\mathbb{S} = \left\{ \vec{p} \in \mathbb{R}^3 \;\middle|\; \vec{p} = \alpha \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \quad \alpha, \beta \in \mathbb{R} \right\}.$$

You know that Robot B is located at $\vec{b} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$. Will you be able to detect Robot B, i.e., **does $\vec{b}$ belong to the subspace $\mathbb{S}$? Show your work and justify your answer.**

(d) (6 points) You are now working on the power source of the robot. The nuclear chain reaction that powers it can be modeled by state transition matrix $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$ and the equation:

$$\vec{v}[n+1] = \mathbf{Q}\vec{v}[n],$$

where $\vec{v}[n]$ is the quantity of reactive species at time step n. $\mathbf{Q}$ is:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 10 \end{bmatrix}.$$

You realize that some components of $\vec{v}[n] \to \infty$ as $n \to \infty$.
To stabilize the system you multiply it by a matrix $\mathbf{R}$:

$$\mathbf{R} = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix}.$$

Now the **new system** would be modeled by

$$\vec{v}[n+1] = \mathbf{QR}\vec{v}[n].$$

Your job is to pick the entries of $\mathbf{R}$, and **you can only use non-negative values in R, so $r_1, r_2, r_3 \geq 0$.**
What are the largest values you can choose for r_1, r_2, and r_3 to ensure that the absolute values of the components in $\vec{v}[n] \not\to \infty$ as $n \to \infty$? Show your work and justify your answer.

(e) (6 points) You want to order a new drone to support your robot fleet. The location of the drone is represented by a vector in $\mathbb{R}^3$. You want to program the drone so that it can travel anywhere in the columnspace of an **invertible** matrix $\mathbf{P} \in \mathbb{R}^{3 \times 3}$. However, a typo in the commands unfortunately means you end up programming the drone to move in the columnspace of $\mathbf{P}^{-1}$. **Is the columnspace of P the same as the columnspace of $\mathbf{P}^{-1}$? Why or why not?**
Hint: Consider the matrix $\mathbf{P}^{-1}$. Is $\mathbf{P}^{-1}$ is invertible?