

UC Berkeley
INFO 206B, Fall 2019
Exam 2

December 9, 2019

Print your name: ______________________________________________________________

• If you need more space to answer a question than we give you, you may use the additional blank sheet of paper attached to your exam. Make sure that we know where to look for your answer.

• Read each question carefully and make sure that you answer everything asked of you. Write legibly so that we can read your solutions. Please do not write anything in red.

• We suggest that for solutions that require you to write Python code, you include comments. They will help your grader understand what you intend, which can help you get partial credit.

• You have until noon to complete the exams
<table>
<thead>
<tr>
<th>Question</th>
<th>Value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Question 1 15 points

Consider the following pair of linear equations.

\[-3y + 2x = -4\]
\[-2y + 3x = -1\]

a.) Solve for \(x\) and \(y\) using any technique that you would like (show all your work).

b.) Formulate these equations as a vector scaling/summation problem.

c.) Sketch (on the next page) the two vectors, their scaled versions, and their vector sum.
**Question 2**  
15 points

Consider the following two linear equations. Formulate these equations in matrix/vector form and then solve for $y$ and $z$ by directly solving this matrix/vector formulation (show all you work).

Clearly denote which quantities in your solution are matrices (capital letters), vectors (lower-case letters with a line above it), and scalars (lower-case letters).

\[
\begin{align*}
4y & = 1 \\
2y & = 1 - 2z
\end{align*}
\]
Question 3 10 points

You are given the 2-D vectors $\vec{x} = (x_1 \ x_2)^T$ and $\vec{y} = (y_1 \ y_2)^T$ and asked to determine a vector $\vec{z} = (z_1 \ z_2)^T$ whose dot-product with $\vec{x}$ and $\vec{y}$ are each 1. Specify the fully constrained linear system (in matrix/vector form) for determining $\vec{z}$. You do not need to solve the system.

Clearly denote which quantities in your solution are matrices (capital letters), vectors (lower-case letters with a line above it), and scalars (lower-case letters).
Question 4 15 points

Consider the following:

\[ a \begin{pmatrix} 5 \\ 2 \\ 1 \end{pmatrix} + b \begin{pmatrix} 15 \\ 6 \\ 3 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \]

a.) Is this an under- or over-constrained system of constraints?

b.) If you believe they exist, give an example of values of \( x, y, \) and \( z \) for which you can solve this equation, otherwise explain why not.

c.) If you believe they exist, give an example of values of \( x, y, \) and \( z \) for which you cannot solve this equation, otherwise explain why not.
Question 5 20 points

You are given \( n > 1 \) points \((x_i, y_i, z_i), \ i = 1, 2, ..., n\), and asked to estimate, using least-squares, the parameter \( a \) that best fit the model: \( z = ax^2 + y \).

a.) Specify the over-constrained system of linear equations in matrix form.

b.) Specify the quadratic-error function for this least-squares estimation.

c.) Solve for the least-squares estimator by differentiating your error function, setting it equal to zero, and then solving for \( a \).

Clearly denote which quantities in your solution are matrices (capital letters), vectors (lower-case letters with a line above it), and scalars (lower-case letters).
Question 6  15 points

You previously wrote a recursive version of binary search. Here you will write an iterative (non-recursive) version of binary search. Your Python function will take two parameters, a list of integers $A$ and an integer $x$. If $x$ is found in the list, then your function should return the index of this element (e.g., if the list is $[1,3,5,7,9]$ and $x$ is 3, then your function will return 1). If $x$ is not found in the list, then your function should return -1. Your function should only use a single while loop and a single if-elif-else ladder.

HINT: the iterative version of binary search works very similar to the recursive version in which you will keep track of a left/right index value, compute the mid-point, determine which half of the list the element (if present) is in, update the values of the left/right indices accordingly, and repeat.
Question 7  
10 points

Below is the code for insertion sort. Using big-O notation, specify the run-time complexity of insertion sort when the list $L$ is sorted, briefly explain your answer?

```python
def insertion_sort(L):
    for i in range (1, len(L)):
        key = L[i]
        j = i-1
        while j >= 0 and L[j] > key:
            L[j+1] = L[j]
            j -= 1
        L[j+1] = key
```