David Lee
SID 3031796951

EE Hw 02

1) William Song 3031799759
Matt Soh 3032109159

~~We~~ First, I worked separately for 4 hours then we met up after EE discussion to talk about how each approached the problems.

2) a) $A^2 = \begin{bmatrix} 0 & 2 & 1 & 3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 2 & 1 & 3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = A^2$

$A^3 = \begin{bmatrix} 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 2 & 1 & 3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = A^3$

$A^4 = \begin{bmatrix} 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 2 & 1 & 3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = A^4$

b) $B^2 = \begin{bmatrix} 3 & 4 & 2 & 1 \\ -5 & -6 & -3 & -1 \\ 6 & 7 & 3 & 2 \\ 2 & 2 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 3 & 4 & 2 & 1 \\ -5 & -6 & -3 & -1 \\ 6 & 7 & 3 & 2 \\ 2 & 2 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 3 & 4 & 1 & 3 \\ -5 & -7 & -2 & -5 \\ 5 & 7 & 2 & 5 \\ 2 & 3 & 1 & 2 \end{bmatrix} = B^2$

$B^3 = \begin{bmatrix} 3 & 4 & 1 & 3 \\ -5 & -7 & -2 & -5 \\ 5 & 7 & 2 & 5 \\ 2 & 3 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} 3 & 4 & 2 & 1 \\ -5 & -6 & -3 & -1 \\ 6 & 7 & 3 & 2 \\ 2 & 2 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ -2 & -2 & 0 & -2 \\ 2 & 2 & 0 & 2 \\ 1 & 1 & 0 & 1 \end{bmatrix} = B^3$

$B^4 = \begin{bmatrix} 1 & 1 & 0 & 1 \\ -2 & -2 & 0 & -2 \\ 2 & 2 & 0 & 2 \\ 1 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 3 & 4 & 2 & 1 \\ -5 & -6 & -3 & -1 \\ 6 & 7 & 3 & 2 \\ 2 & 2 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = B^4$

3) a) Switching rows 1 & 2: $\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Multiplying row 3 by -4: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Adding 2 x row 2 to row 4: & subtracting row 2 from row 1 $\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \end{bmatrix}$

b)

$A = \left[\begin{array}{cccc|c} 1 & -2 & 0 & -5 & 16 \\ 0 & 1 & 0 & 3 & -7 \\ -2 & -3 & 1 & -6 & 9 \\ 0 & 1 & 0 & 2 & -5 \end{array}\right]$

3b Continued

E_1 : Adding 2x row 1 to row 3 $= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$$E_1 A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 0 & -5 \\ 0 & 1 & 0 & 3 \\ -2 & -3 & 1 & -6 \\ 0 & 1 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 0 & -5 \\ 0 & 1 & 0 & 3 \\ 0 & -7 & 1 & -16 \\ 0 & 1 & 0 & 2 \end{bmatrix}$$

E_2 : Adding 7x row 2 to row 3 $= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 7 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$$E_2(E_1 A) = \begin{bmatrix} 1 & -2 & 0 & -5 \\ 0 & 1 & 0 & 3 \\ 0 & -7 & 1 & -16 \\ 0 & 1 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 7 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 0 & -5 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 5 \\ 0 & 1 & 0 & 2 \end{bmatrix}$$

E_3 : Subtracting row 2 from row 4 $= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$

$$E_3(E_2(E_1 A)) = \begin{bmatrix} 1 & -2 & 0 & -5 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 5 \\ 0 & 1 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 0 & -5 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

E_4 : Multiply row 4 by (-1) $= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$, $E_4(E_3(E_2(E_1(A)))) = \begin{bmatrix} 1 & -2 & 0 & -5 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

E_5 : Subtract 5x row 4 from row 3 ~~& 3x row 4 from row 2~~ $= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$$E_5(E_4(E_3(E_2(E_1 A)))) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 0 & -5 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 0 & -5 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

E_6 : Subtract 3x row 4 from row 2 $= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$$E_6(E_5(E_4(E_3(E_2(E_1 A))))) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 0 & -5 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 0 & -5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

E_7 : Add 5x row 4 to row 1 $= \begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $E_7E_6E_5E_4E_3E_2E_1A = \begin{bmatrix} 1 & -2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

E_8 : Add 2x row 2 to row 1 $= \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $E_8E_7E_6E_5E_4E_3E_2E_1A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$\therefore E = E_8(E_7(E_6(E_5(E_4(E_3(E_2 E_1))))))$

$$\boxed{E = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & -2 & 0 & 3 \\ 2 & 2 & 1 & 5 \\ 0 & 1 & 0 & -1 \end{bmatrix}}$$

3b (Continued)

$$EA = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & -2 & 0 & 3 \\ 2 & 2 & 1 & 5 \\ 0 & 1 & 0 & -1 \end{bmatrix} \left[\begin{array}{cccc|c} 1 & -2 & 0 & -5 & 16 \\ 0 & 1 & 0 & 3 & -7 \\ -2 & -3 & 1 & -6 & 9 \\ 0 & 1 & 0 & 2 & -5 \end{array}\right]$$

$$= \left[\begin{array}{cccc|c} 1+0+0+0 & -2+1+0+1 & 0+0+0+0 & -5+3+0+2 & 16-7+0-5 \\ 0+0+0+0 & 0-2+0+3 & 0+0+0+0 & 0-6+0+6 & 0+14+0-15 \\ 2+0-2+0 & -4+2-3+5 & 0+0+1+0 & -10+6-6+10 & 32-14+9-25 \\ 0+0+0+0 & 0+1+0-1 & 0+0+0+0 & 0+3+0-2 & 0-7+0+5 \end{array}\right]$$

$$= \boxed{\left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 & -2 \end{array}\right]}$$

let 1, 2, 3 be A, B, C respectively.

4) a) $\begin{bmatrix} 0 & a & 0 \\ b & 0 & 0 \\ c & 0 & 0 \end{bmatrix} \begin{bmatrix} A_0 \\ B_0 \\ C_0 \end{bmatrix} = \begin{bmatrix} A_1 \\ B_1 \\ C_1 \end{bmatrix} = \begin{bmatrix} aB_0 \\ bA_0 \\ cA_0 \end{bmatrix}$

Corresponding vector-matrix multiplication of the pools vector $\begin{bmatrix} A_0 \\ B_0 \\ C_0 \end{bmatrix}$ and the matrix pump-system $\begin{bmatrix} 0 & a & 0 \\ b & 0 & 0 \\ c & 0 & 0 \end{bmatrix}$, in the original setting.

Thus, $\begin{bmatrix} aB_0 \\ bA_0 \\ cA_0 \end{bmatrix}$ is the resulting amount of fluid in each pool after activating the pumps once.

In the reversed system, the pump matrix is $\begin{bmatrix} 0 & b & c \\ 0 & a & 0 \\ 0 & 0 & 0 \end{bmatrix}$.
Applying the reversed pump system to the initial pools vector $\begin{bmatrix} aB_0 \\ bA_0 \\ cA_0 \end{bmatrix}$ gives us

$$\begin{bmatrix} 0 & b & c \\ 0 & a & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} aB_0 \\ bA_0 \\ cA_0 \end{bmatrix} = \begin{bmatrix} b^2A_0 + c^2A_0 \\ abA_0 \\ 0 \end{bmatrix}$$

4) Next Page for (b) – (e)

let $a = b = c = \frac{1}{3}$; $I_0 = II_0 = III_0 = 1$

a) $\boxed{\text{No}}$, b/c after one time step in the original pump setting, $\begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{2}{3} \\ \frac{1}{3} \end{bmatrix}$

applying one time step of the simple reverse $\begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \frac{2}{3} \\ \text{[illegible]} \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ \frac{2}{9} \\ \text{[illegible]} \end{bmatrix}$ ← does not give us back $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

$\boxed{\text{No}}$

4) b) Taking into account the fluid conservation of the three-pool system, the pumps in original setting can be described with this matrix:

$$M_1 = \begin{bmatrix} 1-(b+c) & 0 & 0 \\ b & 1 & 0 \\ c & 0 & 1 \end{bmatrix},$$ so that each column of the matrix sums to one.

Likewise, the pumps in the reversed setting as a matrix:

$$M_2 = \begin{bmatrix} 1 & b & c \\ 0 & 1-b & 0 \\ 0 & 0 & 1-c \end{bmatrix}$$

Let $b = c = \frac{1}{3}$, the amount of fluid transferred.
M_1 and M_2 become: $M_1 = \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{1}{3} & 0 & 1 \end{bmatrix}$ & $M_2 = \begin{bmatrix} 1 & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{2}{3} & 0 \\ 0 & 0 & \frac{2}{3} \end{bmatrix}$

Represent the pools I, II, and III as a vector: $P_0 = \begin{bmatrix} I_0 \\ II_0 \\ III_0 \end{bmatrix}$,
where I_0, II_0, and III_0 represent the initial amount of fluid in each pool.

If the ~~simple~~ normalized reversion works, $P_1 = M_1 P_0$ and $P_0 = M_2 P_1$.

let $P_0 = \begin{bmatrix} I_0 \\ II_0 \\ III_0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$.

$$P_1 = M_1 P_0 = \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{1}{3} & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ \frac{4}{3} \\ \frac{4}{3} \end{bmatrix}$$

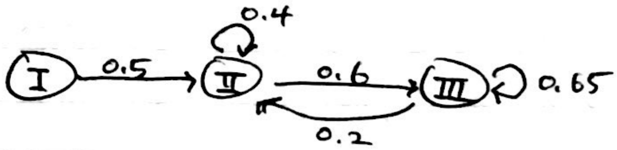
Applying the reversed setting on P_1,

$$P_0 = M_2 P_1 = \begin{bmatrix} 1 & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{2}{3} & 0 \\ 0 & 0 & \frac{2}{3} \end{bmatrix}\begin{bmatrix} \frac{1}{3} \\ \frac{4}{3} \\ \frac{4}{3} \end{bmatrix} = \begin{bmatrix} 11/9 \\ 8/9 \\ 8/9 \end{bmatrix} \neq \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = P_0$$

$\therefore$ The reversion proposed by my fellow intern does not work.

4) c) When the entries of each column vector of a state transition matrix sum to one, the physical interpretation is that the total amount of water in the system is neither mysteriously gained or lost.

d)



$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0.5 & 0.4 & 0.2 \\ 0 & 0.6 & 0.65 \end{bmatrix}$$

After many applictions of the transition matrix A, the water in reservoir I ~~essentially~~ emties out completely while the amount of water in III gets very small to the point where it is almost empty.

e) $A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & a_{ij} & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}$, where the entries of A_{i*} sum to one.

$$A \begin{bmatrix} x \\ \vdots \\ x \end{bmatrix} = \begin{bmatrix} a_{11}x + a_{12}x + \dots a_{1n}x \\ \vdots \\ a_{n1}x + a_{n2}x + \dots a_{nn}x \end{bmatrix}$$

$$= \begin{bmatrix} x(a_{11} + a_{12} + \dots + a_{1n}) \\ \vdots \\ x(a_{n1} + a_{n2} + \dots + a_{nn}) \end{bmatrix} = \begin{bmatrix} x(1) \\ \vdots \\ x(1) \end{bmatrix} = \begin{bmatrix} x \\ \vdots \\ x \end{bmatrix}$$

5) ~~For~~: $k > n$:

No matter what matrix A, any set of vectors in $\mathbb{R}^n$ with more than n elements ($k > n$) must be linearly dependent.

For $k \leq n$:

Identity matrix A assume, then the set

$$\{A\vec{v}_1, A\vec{v}_2, \dots, A\vec{v}_k\}$$ becomes

$$\{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k\},$$

which is given that it is linearly dependent.

Assume zero matrix A, then the set

$$\{A\vec{v}_1, A\vec{v}_2, \dots, A\vec{v}_k\}$$ becomes

$$\{\vec{0}_1, \vec{0}_2, \dots \vec{0}_k\},$$

which is a ~~de~~ linearly dependent set with a zero vector.

$(\vec{0}_k = \alpha_1\vec{0}_1 + \alpha_2\vec{0}_2 + \dots + \alpha_{k-1}\vec{0}_{k-1}$ for any $\alpha_1 \cdots \alpha_{k-1})$

For all other non-Identity, non-zero matrix A given by

$$A_{n \times n} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & a_{ij} & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix};$$

~~Let~~ there exists $\vec{b}$ in $\mathbb{R}^k$ such that $\vec{b} \neq 0$ and $b_1\vec{v}_1 + \dots + b_k\vec{v}_k = 0$. by definition of linear dependence

Then $Ab_1\vec{v}_1 + Ab_2\vec{v}_2 + \dots + Ab_k\vec{v}_k = A(b_1\vec{v}_1 + b_2\vec{v}_2 + \dots + b_k\vec{v}_k)$

$= A(0) = 0$

$\therefore \{A\vec{v}_1, A\vec{v}_2, \dots, A\vec{v}_k\}$ is also linearly dependent

6) a) $P_1, P_2, \ldots P_6$ are constants. given.

The system of equations becomes:

$$
\begin{aligned}
(E_1)&: \quad T_1 + T_2 &&= P_1 \\
(E_2)&: \quad T_2 + T_3 &&= P_2 \\
(E_3)&: \quad T_3 + T_4 &&= P_3 \\
(E_4)&: \quad T_4 + T_5 &&= P_4 \\
(E_5)&: \quad T_5 + T_6 &&= P_5 \\
(E_6)&: \quad T_1 + T_6 &&= P_6
\end{aligned}
\quad \text{or} \quad
\begin{bmatrix} 1&1&0&0&0&0 \\ 0&1&1&0&0&0 \\ 0&0&1&1&0&0 \\ 0&0&0&1&1&0 \\ 0&0&0&0&1&1 \\ 1&0&0&0&0&1 \end{bmatrix}
\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{bmatrix}
=
\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{bmatrix}
$$

We can see that

$$E_6 = E_1 + (-E_2) + E_3 + (-E_4) + E_5,$$ meaning the equations are linearly dependent.

Thus, $\boxed{\text{NO}}$ unique solution can be solved for T_1 to T_6

One example of two different assignments of T_1 to T_6 that result in the same P_1 to P_6 is

$$[T_1, T_2, \ldots, T_6] = [1, 2, 3, 4, 5, 6]$$

and

$$[T_1, T_2, \ldots, T_6] = [2, 1, 4, 3, 6, 5],$$

Which both give $[P_1, P_2, \ldots P_6] = [3, 5, 7, 9, 11, 7]$

b)
$$
\begin{bmatrix} 1&1&0&0&0 \\ 0&1&1&0&0 \\ 0&0&1&1&0 \\ 0&0&0&1&1 \\ 1&0&0&0&1 \end{bmatrix}
\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{bmatrix}
=
\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{bmatrix}
\Rightarrow \text{Augmented} \Downarrow
$$

$$
\left[\begin{array}{ccccc|c} 1&1&0&0&0&P_1 \\ 0&1&1&0&0&P_2 \\ 0&0&1&1&0&P_3 \\ 0&0&0&1&1&P_4 \\ 1&0&0&0&1&P_5 \end{array}\right]
\Rightarrow R_5 - R_1 + R_2 - R_3 + R_4 \Downarrow
$$

$$
\left[\begin{array}{ccccc|c} 1&1&0&0&0&P_1 \\ 0&1&1&0&0&P_2 \\ 0&0&1&1&0&P_3 \\ 0&0&0&1&1&P_4 \\ 0&0&0&0&2&P_5 - P_1 + P_2 - P_3 + P_4 \end{array}\right]
\begin{array}{l} \Rightarrow R_4 - \frac{1}{2}R_5 \\ \Rightarrow R_5/2 \end{array}
\Rightarrow
\left[\begin{array}{ccccc|c} 1&1&0&0&0& \vdots \\ 0&1&1&0&0& \vdots \\ 0&0&1&1&0& \vdots \\ 0&0&0&1&0& \\ 0&0&0&0&1&(P_5 - P_1 + P_2 - P_3 + P_4)/ \end{array}\right]
$$

AND So ON ... Until the left side of the augmented matrix becomes an identity matrix.

Thus, $\boxed{\text{Yes}}$, We can determine the individual tips of all 5 people.

6) c) For all odd n, $n > 2$

7) a) ① $q_x = R_{11} P_x + R_{12} P_y + T_x$

② $q_y = R_{21} P_x + R_{22} P_y + T_y$

Equation ①

Knowns: q_x, P_x, P_y

Unknowns: R_{11}, R_{12}, T_x

Equation ②

Knowns: q_y, P_x, P_y

Unknowns: R_{21}, R_{22}, T_y

There are three unknowns in each equation, so we need three equations for each type (① or ②) for a total of six equations.

To do so, we need 3 pairs of common points $\vec{P}$ & $\vec{q}$

b) $q_1 = P_1$, $q_2 = P_2$, $q_3 = P_3$

$\therefore$

① $q_{1x} = R_{11} P_{1x} + R_{12} P_{1y} + T_x$

② $q_{1y} = R_{21} P_{1x} + R_{22} P_{1y} + T_y$

③ $q_{2x} = R_{11} P_{2x} + R_{12} P_{2y} + T_x$

④ $q_{2y} = R_{21} P_{2x} + R_{22} P_{2y} + T_y$

⑤ $q_{3x} = R_{11} P_{3x} + R_{12} P_{3y} + T_x$

⑥ $q_{3y} = R_{21} P_{3x} + R_{22} P_{3y} + T_y$

d) let P_1, P_2, P_3 co-linear, then $(P_2 - P_1) = k(P_3 - P_1)$ for some $k \in \mathbb{R}$

③ − ① : $q_{2x} - q_{1x} = R_{11}(P_{2x} - P_{1x}) + R_{12}(P_{2y} - P_{1y})$ ⑦

⑤ − ① : $q_{3x} - q_{1x} = R_{11}(P_{3x}\cancel{3} - P_{1x}) + R_{12}(P_{3y} - P_{1y})$ ⑧

Because $(P_2 - 1) = k(P_3 - P_1)$ for P_{nx} and P_{ny} $(n \in 1, 2, 3)$, Eq. ⑧ can be expressed as a multiple of eq. ⑦ and vice versa, meaning these equations represent redundant information. and the system is underdetermined.

7) f) $\begin{bmatrix} q_x \\ q_y \end{bmatrix} = \begin{bmatrix} \lambda\cos\theta & -\lambda\sin\theta \\ \lambda\sin\theta & \lambda\cos\theta \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}$

$= \begin{bmatrix} a & -b \\ b & a \end{bmatrix}$ " + "

$q_x = a p_x - b p_y + T_x$
$q_y = b p_x + a p_y + T_y$ } There are only 4 unknowns, so need 4 equations, Two points.

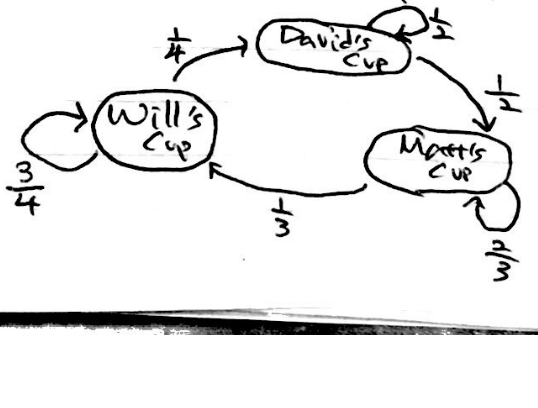
8) Q: Three friends, David, Matt, and William, sit around a table. They decide to play a little game where each poors a certain proportion of ~~th~~ his juice into the person to the left's cup.

They start by pouring themselves unknown amounts of orange juice in their respective cups. Then, David, Matt, and William decide to ~~pour~~ transfer $\frac{1}{2}$, $\frac{1}{3}$, and $\frac{1}{4}$ of their own juice to the person to the left: David to Matt, Matt to William, and William to David.

After this one iteration of juice-transferring, they measured the amount (in liters) of juice each cup held. Magically, each cup had 1 liter of orange juice.

They now want to figure out exactly how much was in each cup before the transfers happened. Can you help them with your EE16A knowledge?

A: Formulate the problem.



let $A = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{4} \\ \frac{1}{2} & \frac{2}{3} & 0 \\ 0 & \frac{1}{3} & \frac{3}{4} \end{bmatrix}$ be the state-transition matrix

let $B_0 = \begin{bmatrix} D_0 \\ M_0 \\ W_0 \end{bmatrix}$ be the initial amounts of juice in each cup.

Thus $AB_0 = B_1$, $B_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

Solve: To solve for $B_0 = A^{-1}B_1$, find the A^{-1}

$$A^{-1}\begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{4} \\ \frac{1}{2} & \frac{2}{3} & 0 \\ 0 & \frac{1}{3} & \frac{3}{4} \end{bmatrix} = I_{3\times 3}$$

Form the augmented matrix:

$$\left[\begin{array}{ccc|ccc} \frac{1}{2} & 0 & \frac{1}{4} & 1 & 0 & 0 \\ \frac{1}{2} & \frac{2}{3} & 0 & 0 & 1 & 0 \\ 0 & \frac{1}{3} & \frac{3}{4} & 0 & 0 & 1 \end{array}\right] \Rightarrow R_2 - R_1$$

$$\left[\begin{array}{ccc|ccc} \frac{1}{2} & 0 & \frac{1}{4} & 1 & 0 & 0 \\ 0 & \frac{2}{3} & -\frac{1}{4} & -1 & 1 & 0 \\ 0 & \frac{1}{3} & \frac{3}{4} & 0 & 0 & 1 \end{array}\right] \Rightarrow R_3 - \frac{R_1}{2} \Rightarrow \left[\begin{array}{ccc|ccc} \frac{1}{2} & 0 & \frac{1}{4} & 1 & 0 & 0 \\ 0 & \frac{2}{3} & -\frac{1}{4} & -1 & 1 & 0 \\ 0 & 0 & \frac{7}{8} & \frac{1}{2} & -\frac{1}{2} & 1 \end{array}\right] \Rightarrow \begin{cases} 2(R_1) \ \& \\ \frac{3}{2}(R_2) \\ \& \\ \frac{8}{7}(R_3) \end{cases}$$

$$\left[\begin{array}{ccc|ccc} 1 & 0 & \frac{1}{2} & 2 & 0 & 0 \\ 0 & 1 & -\frac{3}{8} & -\frac{3}{2} & \frac{3}{2} & 0 \\ 0 & 0 & 1 & \frac{4}{7} & -\frac{4}{7} & \frac{8}{7} \end{array}\right] \Rightarrow \begin{Bmatrix} R_1 - \frac{R_3}{2} \\ R_2 + \frac{3}{8}(R_3) \end{Bmatrix} \Rightarrow \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & \frac{12}{7} & \frac{2}{7} & -\frac{4}{7} \\ 0 & 1 & 0 & -\frac{9}{7} & \frac{9}{4} & \frac{3}{7} \\ 0 & 0 & 1 & \frac{4}{7} & -\frac{4}{7} & \frac{8}{7} \end{array}\right]$$

A^{-1} is the right half of the augmented matrix.

$$\therefore \quad A^{-1} = \begin{bmatrix} \frac{12}{7} & \frac{2}{7} & -\frac{4}{7} \\ -\frac{9}{7} & \frac{9}{7} & \frac{3}{7} \\ \frac{4}{7} & -\frac{4}{7} & \frac{8}{7} \end{bmatrix}$$

Now, to find the initial B_0,

$$B_0 = A^{-1}B = \begin{bmatrix} \frac{12}{7} & \frac{2}{7} & -\frac{4}{7} \\ -\frac{9}{7} & \frac{9}{7} & \frac{3}{7} \\ \frac{4}{7} & -\frac{4}{7} & \frac{8}{7} \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{10}{7} \\ \frac{3}{7} \\ \frac{8}{7} \end{bmatrix} = \begin{bmatrix} D_0 \\ M_0 \\ W_0 \end{bmatrix}$$

Answer:

David originally had $\frac{10}{7}$ liters of O.J.

Matt " " $\frac{3}{7}$ " " "

William " " $\frac{8}{7}$ " " "

Scanned by CamScanner

Problem 7 Image Stitching

This section of the notebook continues the image stiching problem. Be sure to have a `figures` folder in the same directory as the notebook. The `figures` folder should contain the files:

```
Berkeley_banner_1.jpg
Berkeley_banner_2.jpg
stacked_pieces.jpg
lefthalfpic.jpg
righthalfpic.jpg
```

Note: This structure is present in the provided HW2 zip file.

Run the next block of code before proceeding

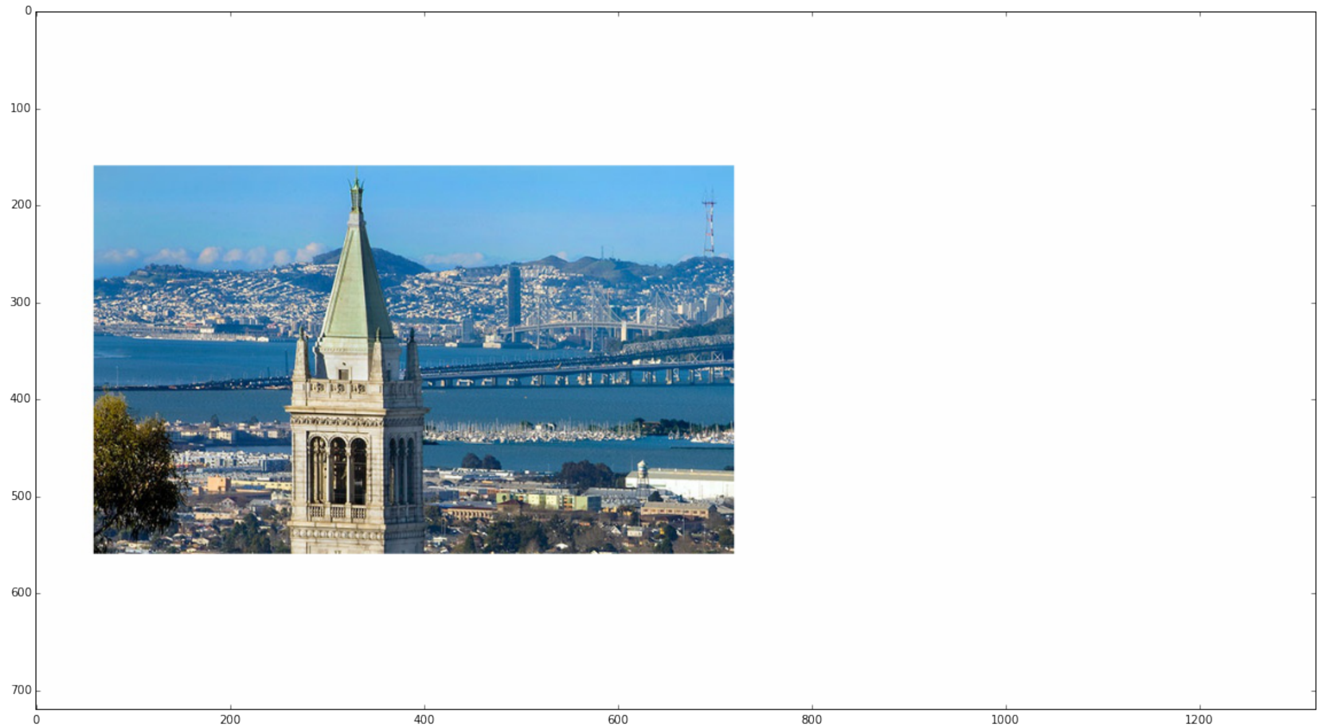
In [1]:
```
import numpy as np
import numpy.matlib
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from numpy import pi, cos, exp, sin
import matplotlib.image as mpimg
import matplotlib.transforms as mtransforms


%matplotlib inline

#loading images
image1=mpimg.imread('figures/Berkeley_banner_1.jpg')
image1=image1/255.0
image2=mpimg.imread('figures/Berkeley_banner_2.jpg')
image2=image2/255.0
image_stack=mpimg.imread('figures/stacked_pieces.jpg')
image_stack=image_stack/255.0


image1_marked=mpimg.imread('figures/lefthalfpic.jpg')
image1_marked=image1_marked/255.0
image2_marked=mpimg.imread('figures/righthalfpic.jpg')
image2_marked=image2_marked/255.0

def euclidean_transform_2to1(transform_mat,translation,image,position,LL,UL):
    new_position=np.round(transform_mat.dot(position)+translation)
    new_position=new_position.astype(int)


    if (new_position>=LL).all() and (new_position<UL).all():
        values=image[new_position[0][0],new_position[1][0],:]
    else:
        values=np.array([2.0,2.0,2.0])

    return values

def euclidean_transform_1to2(transform_mat,translation,image,position,LL,UL):
    transform_mat_inv=np.linalg.inv(transform_mat)
    new_position=np.round(transform_mat_inv.dot(position-translation))
    new_position=new_position.astype(int)

    if (new_position>=LL).all() and (new_position<UL).all():
        values=image[new_position[0][0],new_position[1][0],:]
    else:
        values=np.array([2.0,2.0,2.0])

    return values
```

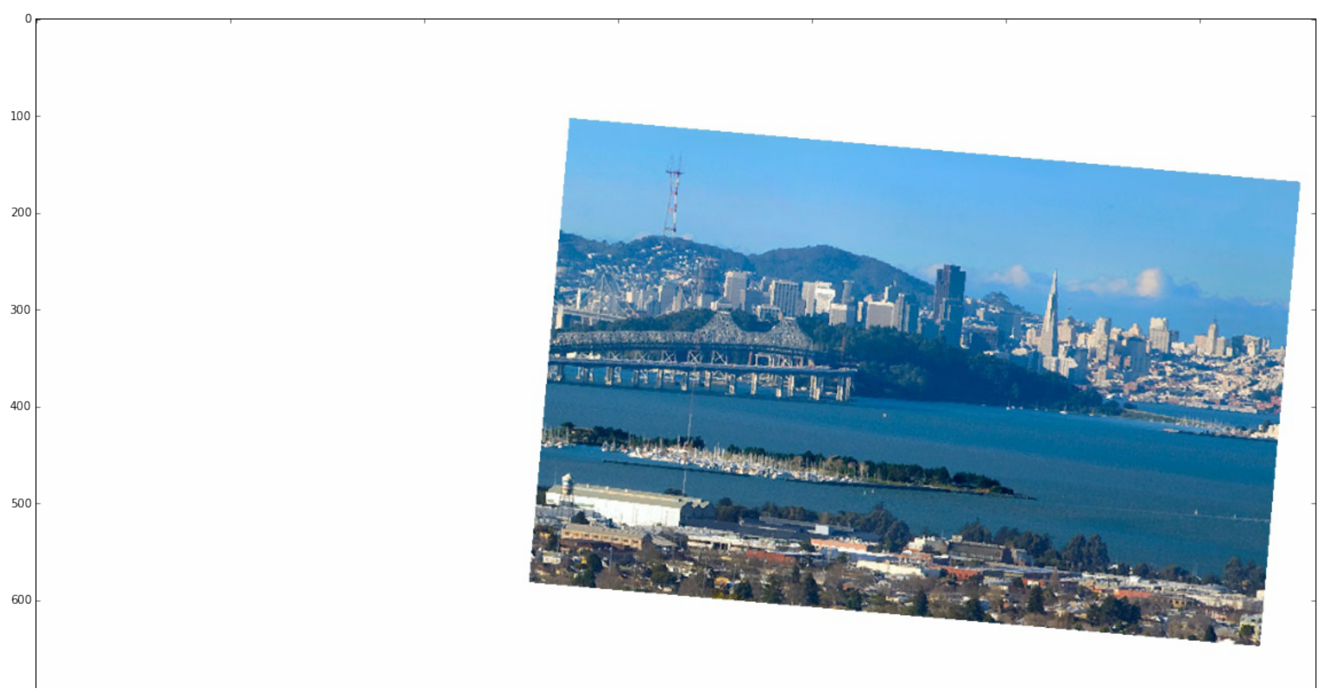
We will stick to a simple example and just consider stitching two images (if you can stitch two pictures, then you could conceivably stitch more by applying the same technique over and over again).

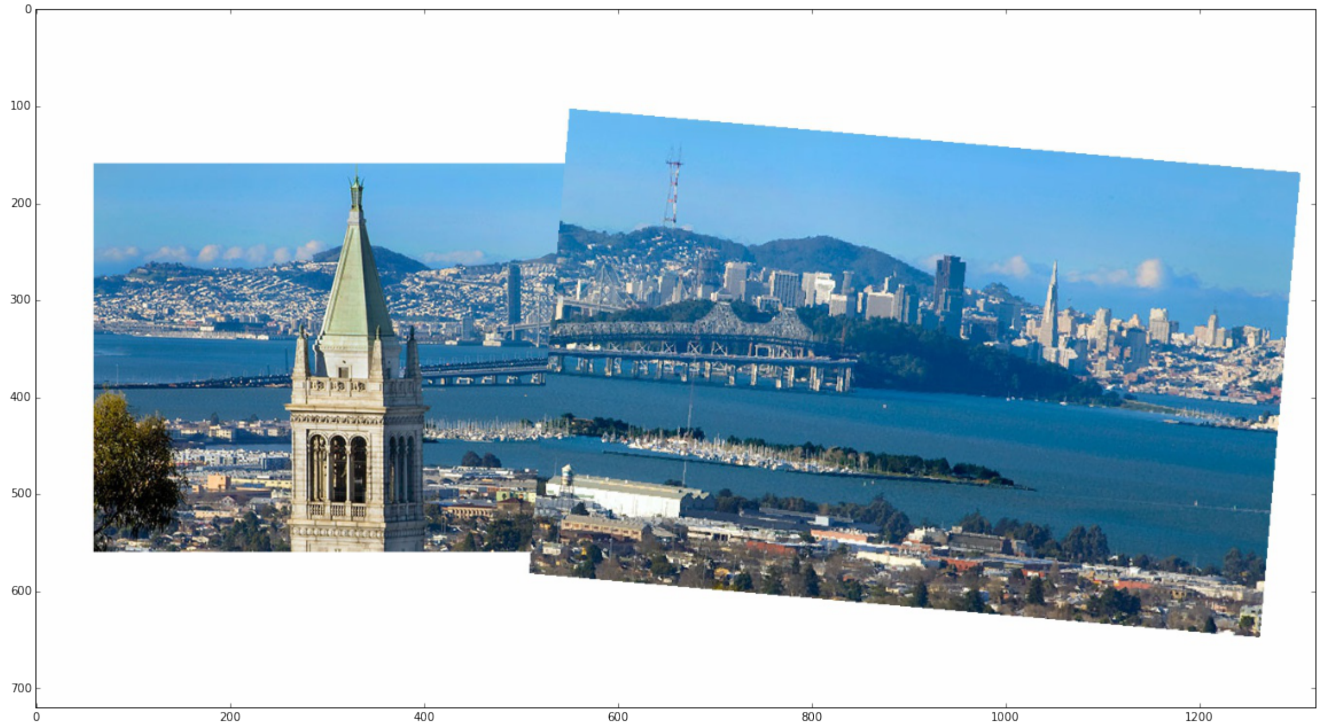
Professor Ayazifar decided to take an amazing picture of the Campanile overlooking the bay. Unfortunately, the field of view of his camera was not large enough to capture the entire scene, so he decided to take two pictues and stich them together.

The next block will display the two images.

In [2]:
```
plt.figure(figsize=(20,40))

plt.subplot(311)
plt.imshow(image1)

plt.subplot(312)
plt.imshow(image2)

plt.subplot(313)
plt.imshow(image_stack)

plt.show()
```



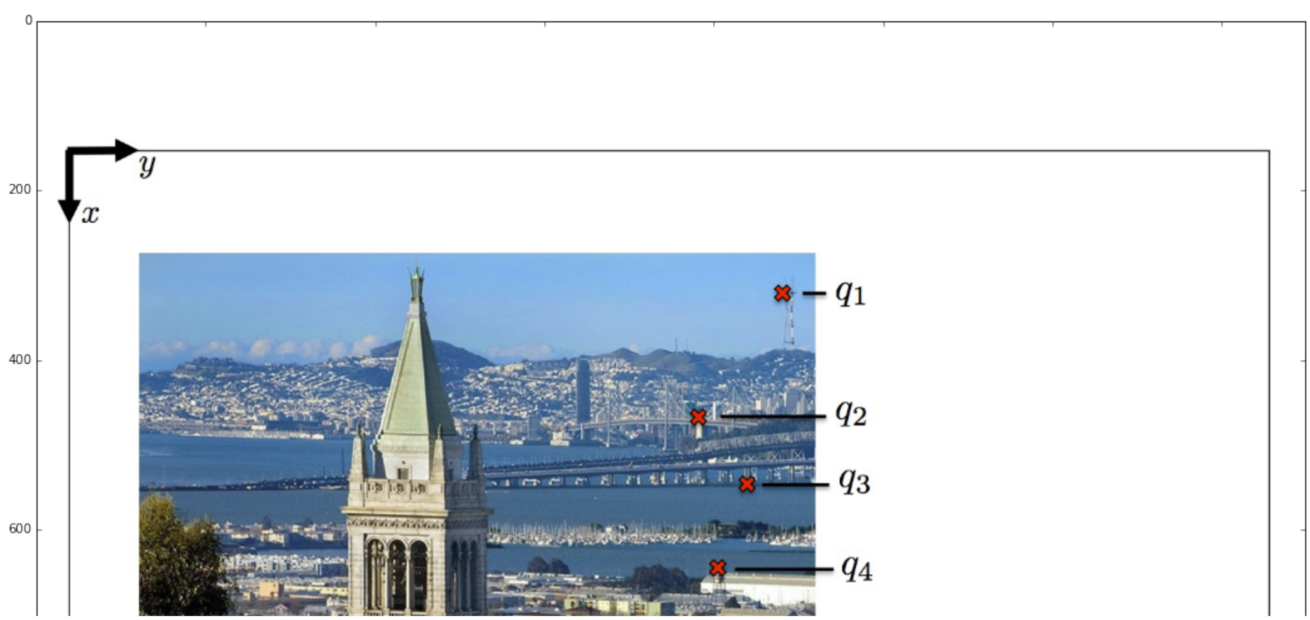


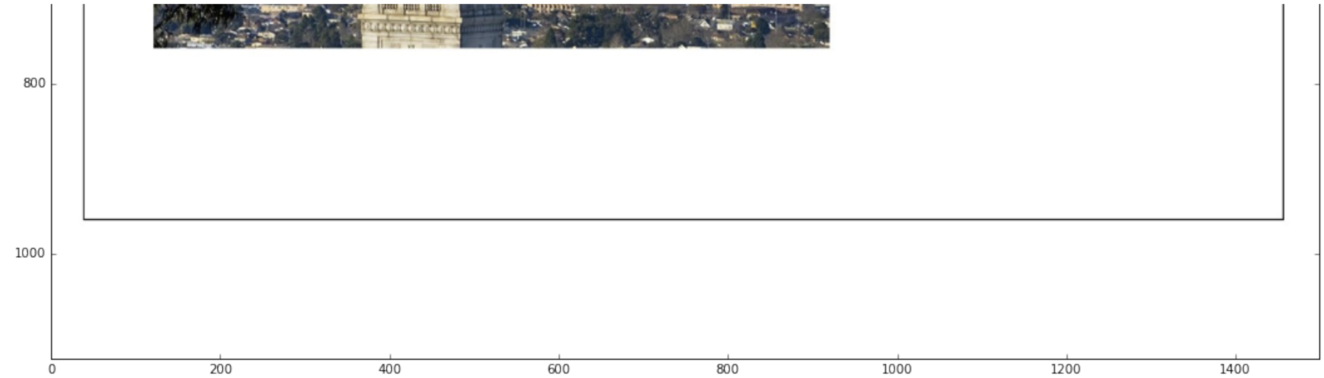


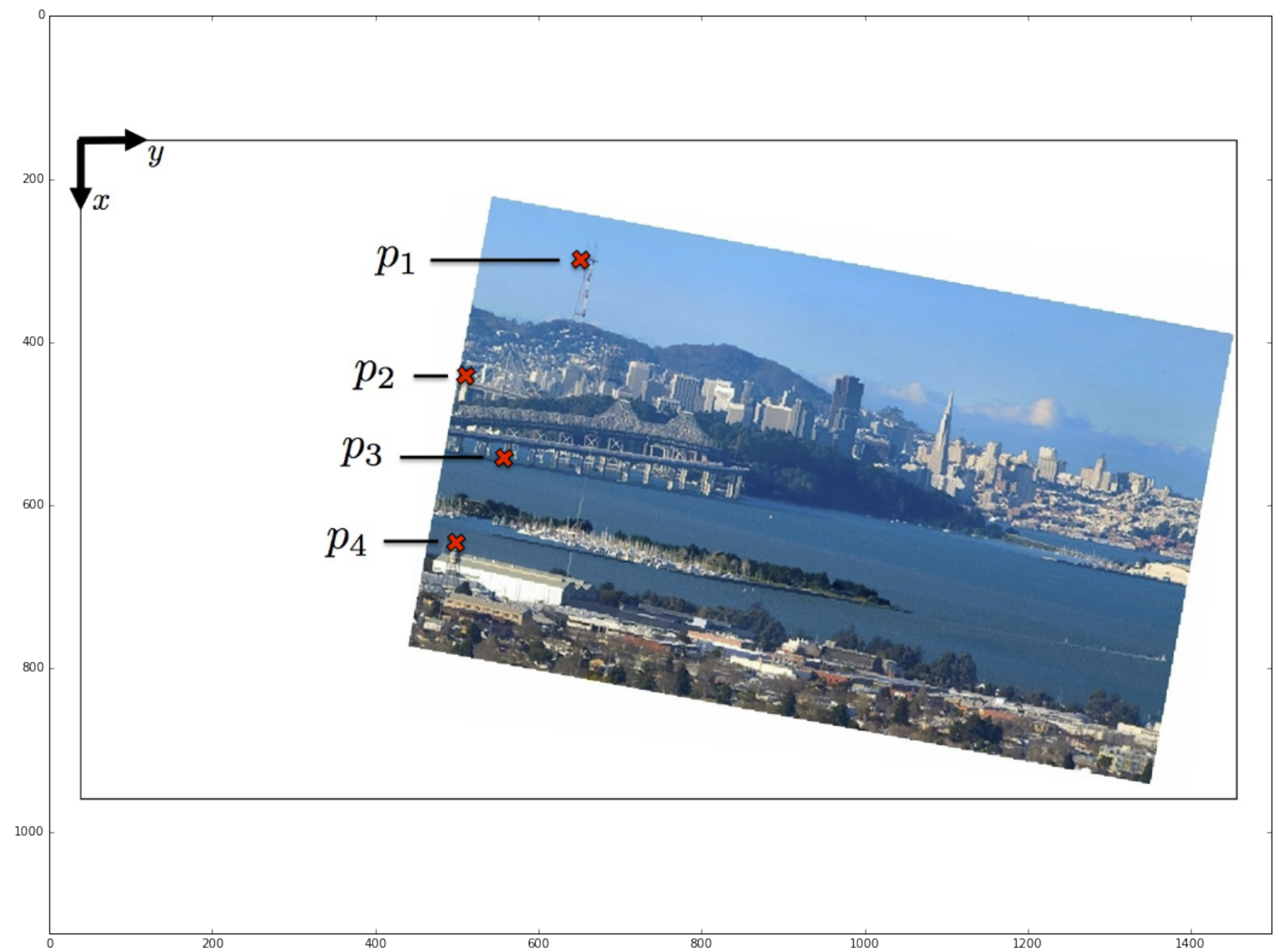
Once you apply Marcela's algorithm on the two images you get the following result (run the next block):

In [3]:
```
plt.figure(figsize=(20,30))

plt.subplot(211)
plt.imshow(image1_marked)

plt.subplot(212)
plt.imshow(image2_marked)
```

Out[3]: <matplotlib.image.AxesImage at 0x105a9a198>







As you can see Marcela's algorithm was able to find four common points between the two images. These points expressed in the coordinates of the first image and second image are

$$\vec{p_1} = \begin{bmatrix} 200 & 700 \end{bmatrix} \quad \vec{p_2} = \begin{bmatrix} 310 & 620 \end{bmatrix} \quad \vec{p_3} = \begin{bmatrix} 390 & 660 \end{bmatrix} \quad \vec{p_4} = \begin{bmatrix} 460 & 630 \end{bmatrix} \quad \vec{q_1} = \begin{bmatrix} 162.2976 & 565.8862 \end{bmatrix} \vec{q_2} = [\, 285.42$$

It should be noted that in relation to the image the positive x-axis is down and the positive y-axis is right. This will have no bearing as to how you solve the problem, however it helps in interpreting what the numbers mean relative to the image you are seeing.

Using the points determine the parameters $R_{11}, R_{12}, R_{21}, R_{22}, T_x, T_y$ that map the points from the first image to the points in the second image by solving an appropriate system of equations. Hint: you do not need all the points to recover the parameters.

In [34]:

```python
# Note that the following is a general template for solving for 6 unknowns from 6 equations represented as Az = b.
# You do not have to use the following code exactly.
# All you need to do is to find parameters R_11, R_12, R_21, R_22, T_x, T_y.
# If you prefer finding them another way it is fine.

# fill in the entries
A = np.array([[200,700,0,0,1,0],
              [0,0,200,700,0,1],
              [310,620,0,0,1,0],
              [0,0,310,620,0,1],
              [390,660,0,0,1,0],
              [0,0,390,660,0,1]])

# fill in the entries
b = np.array([[162.2976],[565.8862],[285.4283],[458.7469],[385.2465],[498.1973]])

A = A.astype(float)
b = b.astype(float)

# solve the linear system for the coefficiens
z = np.linalg.solve(A,b)

#Parameters for our transformation
R_11 = z[0,0]
R_12 = z[1,0]
R_21 = z[2,0]
R_22 = z[3,0]
T_x  = z[4,0]
T_y  = z[5,0]

SUB = str.maketrans("y", "ᵧ")

var = [u'R\u2081\u2081', u'R\u2081\u2082', u'R\u2082\u2081', u'R\u2082\u2082', u'T\u2093', 'Ty'.translate(SUB)]

for i in range(0,6):
    print ('{} = {}'.format(var[i], z[i]))
```

```
R₁₁ = [ 1.1954337]
R₁₂ = [ 0.10458759]
R₂₁ = [-0.10458704]
R₂₂ = [ 1.19543407]
Tₓ = [-150.00045556]
Tᵧ = [-250.00024444]
```

Stitch the images using the transformation you found by running the code below.

Note that it takes about 40 seconds for the block to finish running on a modern laptop.

In [40]:
```python
matrix_transform=np.array([[R_11,R_12],[R_21,R_22]])
translation=np.array([T_x,T_y])

#Creating image canvas (the image will be constructed on this)
num_row,num_col,blah=image1.shape
image_rec=1.0*np.ones((int(num_row),int(num_col),3))

#Reconstructing the original image

LL=np.array([[0],[0]]) #lower limit on image domain
UL=np.array([[num_row],[num_col]]) #upper limit on image domain

for row in range(0,int(num_row)):
    for col in range(0,int(num_col)):
        #notice that the position is in terms of x and y, so the c
        position=np.array([[row],[col]])
        if image1[row,col,0] > 0.995 and image1[row,col,1] > 0.995 and image1[row,col,2] > 0.995:
            temp = euclidean_transform_2to1(matrix_transform,translation,image2,position,LL,UL)
            image_rec[row,col,:] = temp
        else:
            image_rec[row,col,:] = image1[row,col,:]


plt.figure(figsize=(20,20))
plt.imshow(image_rec)
plt.axis('on')
plt.show()
```