Pivoting

The starting point of this tutorial is the LU decomposition code we wrote and tested in lecture 5. We found that it fails when the input matrix has a singular sub matrix, even when the input matrix itself is invertable. For instance

$$A = \begin{pmatrix} 2 & 2 & 1 \\ 2 & 2 & -1 \\ 1 & -1 & 0 \end{pmatrix}$$

has the sub matrix

$$\begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$$

which is singular and therefore our LU decomposition code encounters a division by zero.
The solution to this problem is "pivoting", described in lecture 6. The pseudo-code is on slide 7. Implement this algorithm, following the steps below.

- Write a function that swaps two rows of a matrix. Inputs should be a $n \times n$ array M and indices $0 \leq i, j < n$ (following the Python convention that indices start from 0). Output should be the array M with rows i and j swapped.
- There is a line in the pseudo-code that says "Select $i \geq k$ to maximize $|U_{i,k}|$." Read the documentation on the Argmax function on numpy.org to figure out how to find the index of the largest element in an array. Work out how to use this function to implement the line in the pseudo-code mentioned above.
- Now implement the pseudocode for LU decomposition with pivoting. Input sould be a $n \times n$ array A, output should be $n \times n$ arrays P, L and U so that P is a permutation matrix and $PA = LU$.
- If you have time, test it by decomposing the matrix A above and verifying that $PA = LU$.