1 Solving (Square) Linear System using LU decomposition: Factor and Solve

Let $A \in \mathbb{R}^{n\times n}$. Then there exists an invertible lower triangular matrix $L \in \mathbb{R}^{n\times n}$ with 1's on the diagonals, a matrix $U \in \mathbb{R}^{n\times n}$ in row echelon form and a permutation matrix $P \in \mathbb{R}^{n\times n}$, such that

$$PA = LU.$$

Tasks:

1. **Factorization:** Implement a function `factor_lu(A)` which computes the LU decomposition of a matrix $A \in \mathbb{R}^{n\times n}$. It shall output two `numpdy.ndarray`'s of shape `(n,n)`. One is `lu` which on the lower triangular part (excluding the diagonal) contains the information of L and on the upper triangular part (including the diagonal) contains the information for U. The other is `piv` which is the permutation matrix P.

   Hints:

   - You can recycle previous codes.

2. **Solving:** Implement a function `solve_lu(Lu, P, b)` which takes as input the `numpy.ndarray`'s `Lu` and `P` computed by `factor_lu(A)` as well as a vector $b \in \mathbb{R}^n$. It shall then apply the three-stage solving procedure presented in the lecture and output the solution x of $Ax = b$.

   Hints:

   - Recall the solving procedure:
     1) permute right hand side $\bar{b} = Pb$
     2) solve lower triangular system $Lz = \bar{b}$ for z
     3) solve upper triangular system $Ux = z$ for x
   - For steps 2) and 3) you can use your function `solve_tri(A, b, lower)` from Ex. 2.

3. **Test** your routine on an example.

Solution: