For a regular $m \times n$ matrix $A$, the LU-factorization has the form

$$PA = LU$$

where $P$ is a permutation matrix, $L$ is lower triangular with unit diagonal elements (lower trapezoidal if $m > n$), and $U$ is upper triangular (upper trapezoidal if $m < n$). An implementation can overwrite $A$ compactly with $L$ and $U$ and we therefore denote such an operation by

$$A \leftarrow P^T (L \setminus U) \quad \text{(lu-variant)}.$$ 

Assuming an algorithm $lu\_unblk$ is known we can define algorithm $lu\_blk$ as follows:

- if $\min\{m, n\} < b_{\text{min}} := 2^k$ (for fixed $k \geq 0$)
  
  $$A \leftarrow P (L \setminus U) \quad \text{lu\_unblk}$$

- else choose $bs = \max \{2^\ell : 2^\ell < \min\{m, n\}\}$ and proceed for

  $$A = \begin{pmatrix} A_{0,0} & A_{0,bs} \\ A_{bs,0} & A_{bs,bs} \end{pmatrix}$$

  as follows:

  $$\begin{pmatrix} A_{0,0} \\ A_{bs,0} \end{pmatrix} \leftarrow p^T \begin{pmatrix} L_{0,0} \setminus U_{0,0} \\ L_{bs,0} \end{pmatrix} \quad \text{(lu\_blk)}$$

  $$\begin{pmatrix} A_{0,bs} \\ A_{bs,bs} \end{pmatrix} \leftarrow p \begin{pmatrix} A_{0,bs} \\ A_{bs,bs} \end{pmatrix} \quad \text{(swap)}$$

  $$A_{0,bs} \leftarrow L_{0,0}^{-1} A_{0,bs} \quad \text{(triangular solver)}$$

  $$A_{bs,bs} \leftarrow A_{bs,bs} - A_{bs,0} A_{0,bs} \quad \text{(matrix product)}$$

  $$A_{bs,bs} \leftarrow \tilde{p}^T (L_{bs,bs} \setminus U_{bs,bs}) \quad \text{(lu\_blk)}$$

  $$A_{bs,0} \leftarrow \tilde{p} A_{bs,0} \quad \text{(swap)}$$
Write a function \texttt{lu_blk_var2} that implements this algorithm. You can use the test cases from [session 21](#). But note that your implementation must not call any MKL functions. Only use the corresponding ulmBLAS functions. The file

\texttt{/home/numerik/pub/hpc/ws19/lu_quiz05_stub.hpp}

(available on \textit{theon}) contains a stub for your code.  
Please submit your code “lu.hpp” as follows:

\texttt{theon$ submit hpc quiz05 lu.hpp}