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

- else choose $b_s = \max \{2^\ell : 2^\ell < \min\{m, n\}\}$ and proceed for
  $$A = \begin{pmatrix} A_{0,0} & A_{0,bs} \\ A_{bs,0} & A_{bs,bs} \end{pmatrix}$$

  as follows:

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

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

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

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

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

  $$A_{bs,0} \leftarrow \tilde{P} A_{bs,0} \; \text{(swap)}$$
Write a function `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

/home/numerik/pub/hpc/ws18/lu_quiz05_stub.hpp

(accessible on theon) contains a stub for your code. Please submit your code “lu.hpp” as follows:

theon$ submit hpc quiz05 lu.hpp