Gaussian elimination (without pivoting)

LU-factorization with overwriting $\mathbf{A}$

Input: $\mathbf{A}$

Output: non-trivial entries of $\mathbf{L}$ and $\mathbf{U}$; $\mathbf{A}$ is overwritten by $\mathbf{U}$

for $k = 1 : (n-1)$ do

  for $i = (k+1) : n$ do

    $l_{ik} := \frac{a_{ik}}{a_{kk}}$

    $A(i, [k+1 : n]) += -l_{ik} \cdot A(k, [k+1 : n])$

  end for

end for

Cost: $O(n^3)$ as it involves 3 nested loops

LU-factorization using Crout's method

Input: $\mathbf{A}$, invertible, A has a LU-factorization
Output: algorithm replaces a_{ij} with u_{ij} for $j \geq i$ and with l_{ij} for $j < i$

for $i = 1 : n$ do
  for $k = i : n$ do
    $$a_{ik} := a_{ik} - \sum_{j=1}^{i-1} a_{ij}\, a_{jk}$$
  end for
  for $k = (i+1) : n$ do
    $$a_{ki} := \left(a_{ki} - \sum_{j=1}^{i-1} a_{kj}\, a_{ji} \right) \Big/ a_{ii}$$
  end for
end for

Cost: $O(n^3)$ as it involves $\mathbf{3}$ nested loops