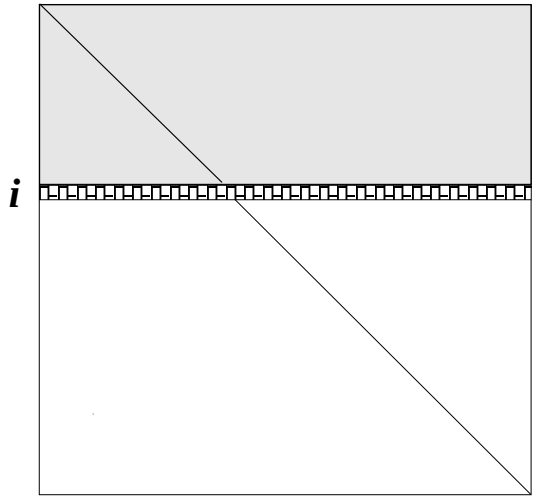
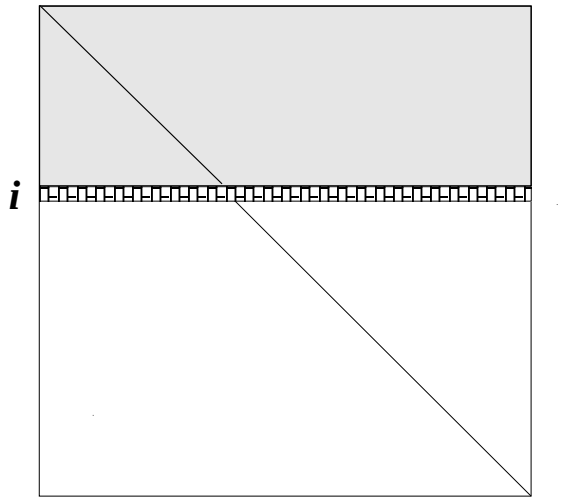
Algorithm 1 **ikj** *lu decomposition (delayed row dense algorithm)*

$l = I_n$
$u = O_n$
$u_{11:n} = a_{1,1:n}$
for i=2:n
  for k=1:i-1
    $l_{ik} = a_{ik}/a_{kk}$
    for j=k+1:n
      $a_{ij} = a_{ij} - l_{ik} * a_{kj}$
    end
  end
  $u_{ii:n} = a_{ii:n}$
end



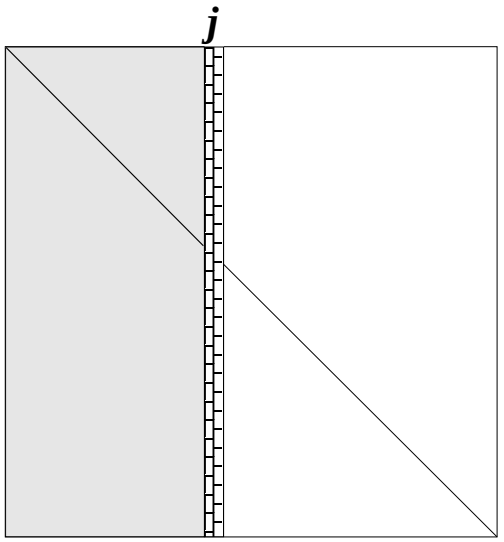
Algorithm 2 ijk *lu decomposition (dot product - based row dense algorithm)*

$l = I_n$, $u = O_n$, $u_{11:n} = a_{11:n}$
for i=2:n
  for j=2:i
    $l_{ij-1} = a_{ij-1}/a_{j-1j-1}$
    for k=1:j-1
      $a_{ij} = a_{ij} - l_{ik} * a_{kj}$
    end
  end
  for j=i+1:n
    for k=1:i-1
      $a_{ij} = a_{ij} - l_{ik} * a_{kj}$
    end
  end
  $u_{i,i:n} = a_{i,i:n}$
end



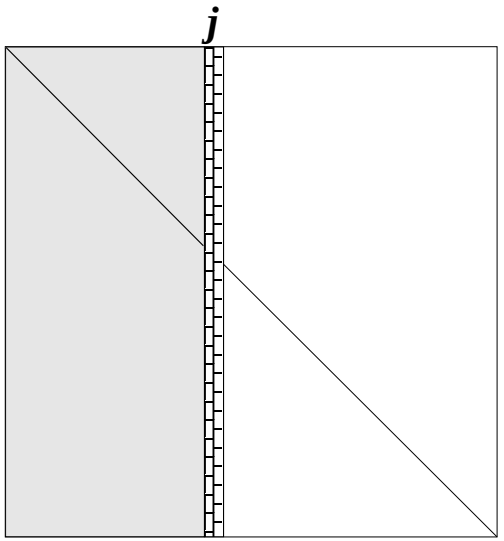
Algorithm 3 jki *lu decomposition (delayed column dense algorithm)*

$l = I_n$, $u = O_n$, $u_{11} = a_{11}$
for j=2:n
  for s=j:n
    $l_{sj-1} = a_{sj-1}/a_{j-1j-1}$
  end
  for k=1:j-1
    for i=k+1:n
      $a_{ij} = a_{ij} - l_{ik} * a_{kj}$
    end
  end
  $u_{1:jj} = a_{1:jj}$
end



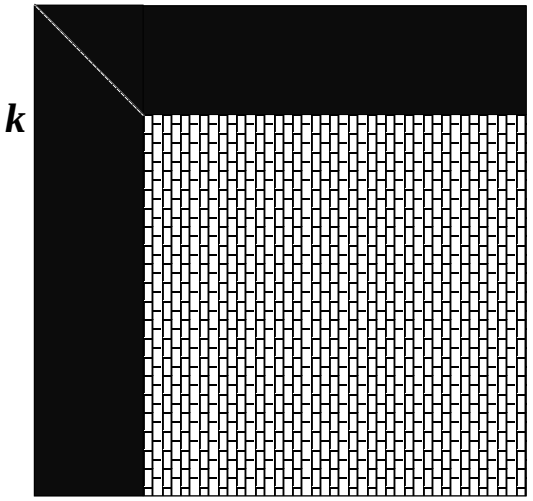
Algorithm 4 jik *lu decomposition (dot product - based column dense algorithm)*

$l = I_n$, $u_{11} = a_{11}$
for j=2:n
  for s=j:n
    $l_{sj-1} = a_{sj-1}/a_{j-1j-1}$
  end
  for i=2:j
    for k=1:i-1
      $a_{ij} = a_{ij} - l_{ik} * a_{kj}$
    end
  end
  for i=j+1:n
    for k=1:j-1
      $a_{ij} = a_{ij} - l_{ik} * a_{kj}$
    end
  end
  $u_{1:jj} = a_{1:jj}$
end



Algorithm 5 kij *lu decomposition (row oriented submatrix dense algorithm)*

$l = I_n$
$u = O_n$
for k=1:n-1
  for i=k+1:n
    $l_{ik} = a_{ik}/a_{kk}$
    for j=k+1:n
      $a_{ij} = a_{ij} - l_{ik} * a_{kj}$
    end
  end
  $u_{kk:n} = a_{kk:n}$
end
$u_{nn} = a_{nn}$



Algorithm 6 kji *lu decomposition (column oriented submatrix dense algorithm)*

$l = I_n$, $u = O_n$

for k=1:n-1

  for s=k+1:n

    $l_{sk} = a_{s,k}/a_{k,k}$

  end

  for j=k+1:n

    for i=k+1:n

      $a_{ij} = a_{ij} - l_{ik} * a_{kj}$

    end

  end

  $u_{kk:n} = a_{kk:n}$

end

$u_{nn} = a_{nn}$