Algorithm 1 Gaussian elimination algorithm with partial pivoting

```
1: procedure GaussPivot(N, A, b, x)          ▷ Solves linear system Ax = b
2:     for k = 1 to N step 1 do                          ▷ Forward reduction
3:         PivotSwap(N, A, b, k)                         ▷ Call PivotSwap
4:         if A_{k,k} = 0 then
5:             STOP                                  ▷ Matrix A is not invertible!
6:         end if
7:         for i = k + 1 to N step 1 do
8:             b_i ← b_i − A_{i,k} b_k / A_{k,k}
9:             for j = k + 1 to N step 1 do
10:                A_{i,j} ← A_{i,j} − A_{i,k} A_{k,j} / A_{k,k}
11:            end for
12:            A_{i,k} ← 0
13:        end for
14:    end for
15:    for i = N to 1 step −1 do                         ▷ Backward substitution
16:        s ← 0                                         ▷ Initialize the sum
17:        for j = i + 1 to N step 1 do
18:            s ← s + A_{i,j} x_j
19:        end for
20:        x_i ← (b_i − s) / A_{i,i}
21:    end for
22:    return x                                          ▷ Return the solution
23: end procedure
```

```
1: procedure PivotSwap(N, A, b, k)                ▷ Swaps the pivot if needed
2:     A_max ← A_{k,k}
3:     p ← k                                      ▷ Pivot row index
4:     for i = k + 1 to N step 1 do               ▷ Find the maximum pivot
5:         if |A_{i,k}| > |A_max| then
6:             A_max ← A_{i,k}
7:             p ← i
8:         end if
9:     end for
10:    if p ≠ k then                              ▷ Swap rows
11:        D ← b_p                                ▷ Dummy variable D
12:        b_p ← b_k
13:        b_k ← D
14:        for j = 1 to N step 1 do
15:            D ← A_{p,j}
16:            A_{p,j} ← A_{k,j}
17:            A_{k,j} ← D
18:        end for
19:    end if
20:    return A, b                                ▷ Return the modified matrices
21: end procedure
```