Example:

$A = LU$

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 1 & 0 & 1 \end{bmatrix}$$

eliminate below the

fix this by swaping rows first

$r_2 \leftarrow r_2 - 4r_1$ (row, col, put 4 there)

$r_2 \leftrightarrow r_1$

by swapping rows so the pivot has the largest magnitude, errors are kept under control.

pivot

$$\begin{bmatrix} 4 & 5 & 6 \\ 1 & 2 & 3 \\ 1 & 0 & 1 \end{bmatrix}$$

$r_2 \leftarrow r_2 - \frac{1}{4} r_1$

$r_3 \leftarrow r_3 - \frac{1}{4} r_1$

$$\begin{bmatrix} 4 & 5 & 6 \\ 0 & 3/4 & 3/2 \\ 0 & -5/4 & -1/2 \end{bmatrix}$$

largest magnitude among the possible pivots

Note, we are working with fractions so no rounding error. So this isn't needed and only an inconvenience... However with floating point it's a necessity...

$r_2 \leftrightarrow r_3$

$$\begin{bmatrix} 4 & 5 & 6 \\ 0 & -5/4 & -1/2 \\ 0 & 3/4 & 3/2 \end{bmatrix}$$

$r_3 \leftarrow r_3 + \frac{3}{5} r_2$

$$U = \begin{bmatrix} 4 & 5 & 6 \\ 0 & -5/4 & -1/2 \\ 0 & 0 & 6/5 \end{bmatrix}$$

```
julia> Ar=[4 5 6; 0 -5//4 -1//2; 0 3//4 3//2]
3×3 Matrix{Rational{Int64}}:
 4//1   5//1   6//1
 0//1  -5//4  -1//2
 0//1   3//4   3//2

julia> Ar[3,:]=Ar[3,:]+3//5*Ar[2,:];

julia> Ar
3×3 Matrix{Rational{Int64}}:
 4//1   5//1   6//1
 0//1  -5//4  -1//2
 0//1   0//1   6//5
```

```
julia> using LinearAlgebra

julia> A=[1 2 3; 4 5 6; 1 0 1]
3×3 Matrix{Int64}:
 1  2  3
 4  5  6
 1  0  1

julia> luans=lu(A)
LU{Float64, Matrix{Float64}}
L factor:
3×3 Matrix{Float64}:
 1.0    0.0  0.0
 0.25   1.0  0.0
 0.25  -0.6  1.0
U factor:
3×3 Matrix{Float64}:
 4.0   5.0    6.0
 0.0  -1.25  -0.5
 0.0   0.0    1.2
```

Built in LU factorization gives the same answer

all these things are now stored in luans

Component	Description
F.L	L (lower triangular) part of LU
F.U	U (upper triangular) part of LU
F.p	(right) permutation Vector
F.P	(right) permutation Matrix

```
julia> luans.U
3×3 Matrix{Float64}:
 4.0   5.0    6.0
 0.0  -1.25  -0.5
 0.0   0.0    1.2

julia> luans.L
3×3 Matrix{Float64}:
 1.0    0.0  0.0
 0.25   1.0  0.0
 0.25  -0.6  1.0

julia> luans.L*luans.U
3×3 Matrix{Float64}:
 4.0  5.0  6.0
 1.0  0.0  1.0
 1.0  2.0  3.0
```

These were the row swaps

$r_1 \leftrightarrow r_2$

$r_2 \leftrightarrow r_3$

$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 1 & 0 & 1 \end{bmatrix}$

$r_1 \leftrightarrow r_2$

456
123
101

$r_2 \leftrightarrow r_3$

456
100
123

permutations of the rows in A.

The permutation p tells how the rows are reorganized..

```
julia> luans.p
3-element Vector{Int64}:
 2
 3
 1
```

p[1] = 2 the 2nd row is now in the 1st place
p[2] = 3 the 3rd row is now in the 2nd place
p[3] = 1 the 1st row is now in the 3rd place