# PLU 分解 与矩阵求逆

## 函数主体内容

### PLU分解

PLU分解过程主要包含了几个基本计算内容：

- 寻找主元并交换行（同时交换A, L, P矩阵，其中P矩阵初始化为单位矩阵）
- 根据找到的主元，消去该列的其他行，并设置L 矩阵该行的内容
- 根据原矩阵的行和得到的L 矩阵的列，求出消去的A矩阵
- 重复上述过程，将A矩阵变为U矩阵
- 最终返回矩阵P , L , U

### 矩阵求逆

根据分析，有

$$
\begin{gathered}
PA = LU \\
A^{-1} = (P^{-1}LU)^{-1} \\
A^{-1} = U^{-1}L^{-1}P
\end{gathered}
$$

所以需要对L和U矩阵进行求逆，并进行矩阵乘法来得到最终的A矩阵的逆矩阵。

在这里进行一个优化，由于L 是下三角矩阵，而U 是上三角矩阵，对这两个矩阵利用高斯消去法进行求逆的过程中，能够有一个简化的过程。

这里以下三角矩阵为例，可以直接从第一行开始利用高斯消去法逐行进行消去。同理，上三角矩阵也可。

### 矩阵乘法

矩阵乘法的逻辑是非常简单的，这里直接采用最基本的矩阵乘法的方式进行求解。

## 输入示例

### 1.

$$
\begin{bmatrix}
4 & 0 & 4 \\
6 & 0 & 3 \\
3 & 7 & 5
\end{bmatrix}
$$

**2.**

$$\begin{bmatrix} 1 & 2 & 4 & 17 \\ 3 & 6 & -12 & 3 \\ 2 & 3 & -3 & 2 \\ 0 & 2 & -2 & 6 \end{bmatrix}$$

## 输出示例

**1.**

$$\begin{bmatrix} -0.25 & 0.33 & 0 \\ -0.25 & 0.095 & 0.14 \\ 0.5 & -0.33 & 0 \end{bmatrix}$$

**2.**

$$\begin{bmatrix} 0.35 & 0.35 & -0.2 & -1.1 \\ -0.375 & -0.542 & 1 & 1 \\ -0.075 & -0.242 & 0.4 & 0.2 \\ 0.1 & 0.1 & -0.2 & 0.1 \end{bmatrix}$$

## 程序正确性比对

这里利用了自己实现的求逆算法和python numpy 中自带的算法进行了比较，由于存在一定的精度误差，故这里的比较条件为

$$sum(Ainv - M) < 0.1$$

其中$Ainv$代表我们所求出来的$A$的逆矩阵，而$M$表示用numpy自带方法求出来的逆矩阵。最终经过多次试验，表明了本程序的正确性。