LU求逆矩阵

import numpy  
def qiuni(A):  
 n = len(A[0])  
 L = numpy.zeros([n, n]) #用零填充的数组  
 U = numpy.zeros([n, n])  
 for i in range(n):  
 L[i][i] = 1  
 if i == 0:  
 U[0][0] = A[0][0]  
 for j in range(1, n):  
 U[0][j] = A[0][j]  
 L[j][0] = A[j][0] / U[0][0]  
 else:  
 for j in range(i, n):  
 val = 0  
 for k in range(0, i):  
 val = val + L[i][k] \* U[k][j]  
 U[i][j] = A[i][j] - val  
 for j in range(i + 1, n):  
 val = 0  
 for k in range(0, i):  
 val = val + L[j][k] \* U[k][i]  
 L[j][i] = (A[j][i] - val) / U[i][i]  
 return L, U  
  
A = [[1, 2, 2], [2, 2, 4], [3, 4, 2]]  
L, U = qiuni(A)  
Res = U \* L  
print(numpy.mat(A))  
print(L)  
print(U)  
print(Res)

