MATH 5610 HW7 SOLUTIONS

Problem 1 Take $A = \begin{bmatrix} 1 & 3 \\ 3 & 1 \end{bmatrix}$, $A = A^T$ and all entries are > 0.

However $\begin{bmatrix} 1 & -1 \end{bmatrix} A \begin{bmatrix} 1 \\ -1 \end{bmatrix} = -4$

Problem 2

$$A = \begin{bmatrix} 4 & 4 & -2 \\ 4 & 13 & 1 \\ -2 & 1 & 18 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 4 & -2 \\ 0 & 9 & 3 \\ 0 & 3 & 17 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 2 & -1 \\ 0 & 9 & 3 \\ 0 & 3 & 17 \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} 2 & 2 & -1 \\ 0 & 9 & 3 \\ 0 & 0 & 16 \end{bmatrix} \leftarrow \begin{bmatrix} 2 & 2 & -1 \\ 0 & 3 & 1 \\ 0 & 0 & 16 \end{bmatrix} \leftarrow \begin{bmatrix} 2 & 2 & -1 \\ 0 & 3 & 1 \\ 0 & 0 & 4 \end{bmatrix} = U$$

can check $A = U^T U$

Problem 3: please see attached code and output.

```
>> test_mylu_sol

A =

     0     2    -1
     1    -1     2
     1    -1     4


L =

     1     0     0
     0     1     0
     1     0     1


U =

     1    -1     2
     0     2    -1
     0     0     2


P =

     0     1     0
     1     0     0
     0     0     1

error=0

A =

     1     1    -1     2
    -1    -1     1     5
     2     2     3     7
     2     3     4     5


L =

    1.0000         0         0         0
    1.0000    1.0000         0         0
    0.5000         0    1.0000         0
   -0.5000         0   -1.0000    1.0000


U =

    2.0000    2.0000    3.0000    7.0000
         0    1.0000    1.0000   -2.0000
         0         0   -2.5000   -1.5000
         0         0         0    7.0000


P =

     0     0     1     0
     0     0     0     1
     1     0     0     0
     0     1     0     0

error=0
```



```
% B&F 6.5.4 a
A = [ 0  2 -1
      1 -1  2
      1 -1  4]
[L,U,P]=mylu(A)
fprintf('error=%g\n',norm(P*A - L*U,'inf'));

% B&F 6.5.4 c
A = [ 1  1 -1 2
     -1 -1  1 5
      2  2  3 7
      2  3  4 5]
[L,U,P]=mylu(A)
fprintf('error=%g\n',norm(P*A - L*U,'inf'));
```

```
function [L,U,P] = mylu(A)
n = size(A,1);
U=A; L=eye(n); P=eye(n);
for k=1:n-1,
 % choose pivot
 [val,i] = max(abs(U(k:n,k)));
 i = i+k-1;
 % switch k-th row with pivot row i
 U([i,k],k:n) = U([k,i],k:n);
 % switch rows in L
 L([i,k],1:k-1)=L([k,i],1:k-1);
 % switch rows in P
 P([i,k],:) = P([k,i],:);
 % update
 for j=k+1:n,
  L(j,k) = U(j,k)/U(k,k);
  U(j,k:n) = U(j,k:n) - L(j,k)*U(k,k:n);
 end;
end;
```