Homework2

CSCI-6351
Wen Liu

#Problem1.

a. For string = “ababbccab”.

The output (y, L) as follow:

```
y =

  9×1 char 数组

    'c'
    'b'
    'b'
    'a'
    'a'
    'a'
    'b'
    'c'
    'b'
```

y =

L = 2

For string = “the dog in the fog”. We see # as the white space. The output (y,L) as follow:

```
y =

  18×1 char 数组

    'e'
    'e'
    'g'
    'n'
    '#'
    'h'
    'h'
    '#'
    'o'
    'o'
    't'
    't'
    '#'
    'i'
    'd'
    'f'
    'g'
    '#'
```

y =

L = 17

b.

```
function B = inverseBWT(y)
    %get the length of y
    [n,] = size(y);
    B = y;

    for i = 1:n-1
        % sort the y
        tmp = sortrows(B);

        % get the last column of sorted newY
        last_column = tmp(1:n, i);

        % append the last column to the exist y
        B(1:n, i+1) = last_column;
    end

    %finally sort the y, we can get the B
    B = sortrows(B);
end
```

c.

```
function x = getOriginalX(y, L)
    % get the matrix of B
    B = inverseBWT(y);
    % get the orginal x from B
    x = B(L, :);
end
```

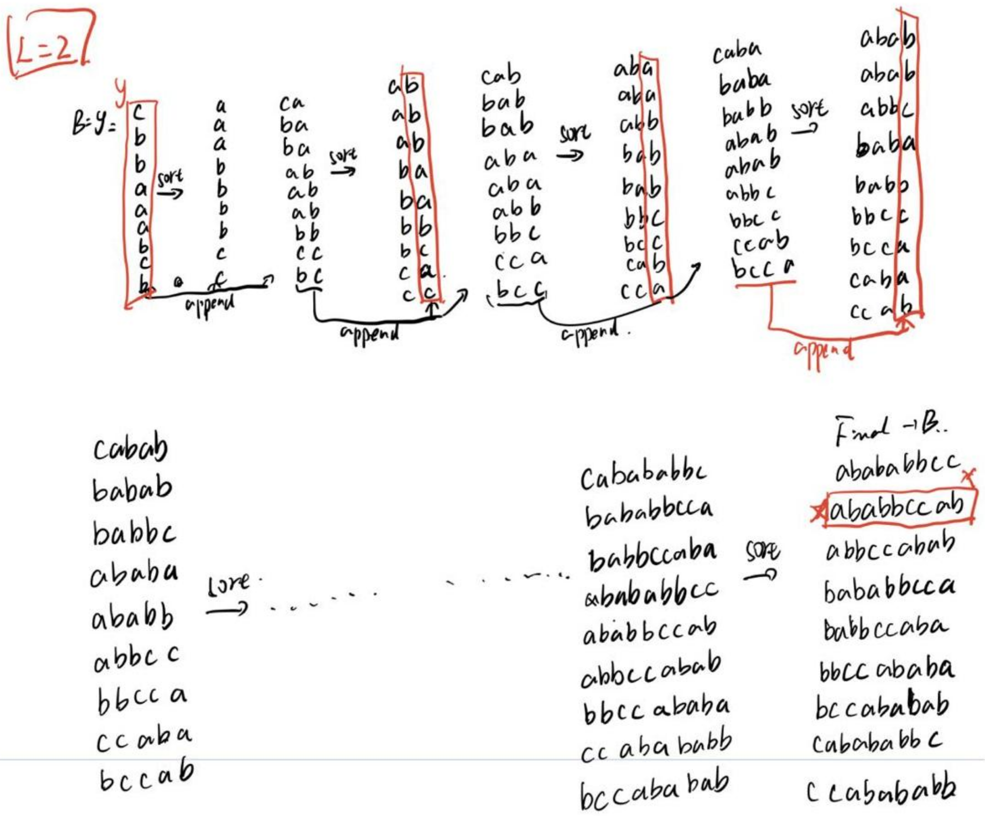
d.

After BWT, we get the (y,L) where y is the last column of B, and L is the location of the original x in B; Therefore we just need to reconstruct the matrix of B from y. For original x, use the B(L) to return the x.

Construct the matrix of B from y: We know the y is the last column of B, and B is the rows of A lexicographically.

Pseudocode code:

```
    B = y
    For i = 1:length(y) - 1
        1.  Sort the B and get last column of B.
            (we use a temporary value to record the sorted B.)
        2.  append the last column of B to the exist B
    B = sort(B).
End.
```



#Problem2.

All the variables means:

Xfft is the Fourier transform of x, *Yfft* is similar.

Xabs is the magnitudes of the elements of Xfft, Yasb is similar

XHat is a column derived from *Xfft* by replacing each of the 11 smallest-magnitude elements of *Xfft* by 0, and leaving the other elements intact. *YHat* is defined similarly.

xHat is the inverse Fourier transform of *XHat*, *yHat* is similar.

a.

```
table1 = [Xfft' Xabs' XHat'];
```

```
>> display(table1)

table1 =

   86.0000   86.0000   86.0000
   52.5483   52.5483   52.5483
   13.6569   13.6569   13.6569
    6.4797    6.4797         0
    4.0000    4.0000         0
    2.8929    2.8929         0
    2.3431    2.3431         0
    2.0791    2.0791         0
    2.0000    2.0000         0
    2.0791    2.0791         0
    2.3431    2.3431         0
    2.8929    2.8929         0
    4.0000    4.0000         0
    6.4797    6.4797         0
   13.6569   13.6569   13.6569
   52.5483   52.5483   52.5483
```

```
table2 = [Yfft' Yabs' YHat'];
```

```
>> display(table2)

table2 =

  -8.0033 + 0.0000i   8.0033 + 0.0000i  -8.0033 + 0.0000i
   4.9858 +11.4797i  12.5157 + 0.0000i   4.9858 +11.4797i
   1.1401 + 1.3960i   1.8024 + 0.0000i   1.1401 + 1.3960i
   0.8148 + 0.8346i   1.1664 + 0.0000i   0.0000 + 0.0000i
   0.7077 + 0.5509i   0.8968 + 0.0000i   0.0000 + 0.0000i
   0.6607 + 0.3661i   0.7554 + 0.0000i   0.0000 + 0.0000i
   0.6376 + 0.2263i   0.6765 + 0.0000i   0.0000 + 0.0000i
   0.6265 + 0.1086i   0.6358 + 0.0000i   0.0000 + 0.0000i
   0.6232 + 0.0000i   0.6232 + 0.0000i   0.0000 + 0.0000i
   0.6265 - 0.1086i   0.6358 + 0.0000i   0.0000 + 0.0000i
   0.6376 - 0.2263i   0.6765 + 0.0000i   0.0000 + 0.0000i
   0.6607 - 0.3661i   0.7554 + 0.0000i   0.0000 + 0.0000i
   0.7077 - 0.5509i   0.8968 + 0.0000i   0.0000 + 0.0000i
   0.8148 - 0.8346i   1.1664 + 0.0000i   0.0000 + 0.0000i
   1.1401 - 1.3960i   1.8024 + 0.0000i   1.1401 - 1.3960i
   4.9858 -11.4797i  12.5157 + 0.0000i   4.9858 -11.4797i
```

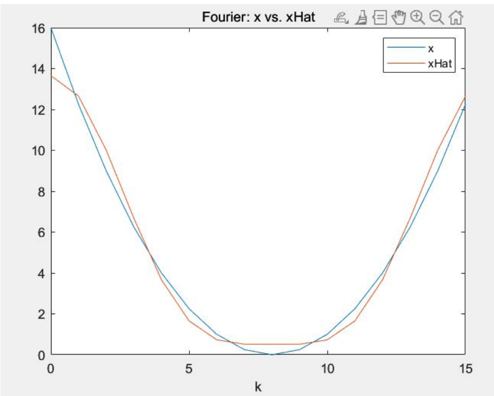
```
table3 = [x' xHat'];
```

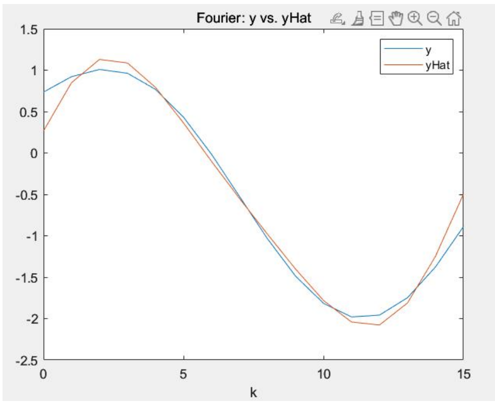
```
>> display(table3)

table3 =

   16.0000   13.6506
   12.2500   12.6506
    9.0000   10.0197
    6.2500    6.6816
    4.0000    3.6679
    2.2500    1.6542
    1.0000    0.7303
    0.2500    0.5136
         0    0.5136
    0.2500    0.5136
    1.0000    0.7303
    2.2500    1.6542
    4.0000    3.6679
    6.2500    6.6816
    9.0000   10.0197
   12.2500   12.6506
```

```
table4 = [y' yHat'];
```

```
>> display(table4)

table4 =

    0.7354    0.2655
    0.9213    0.8489
    1.0086    1.1297
    0.9625    1.0866
    0.7678    0.7922
    0.4320    0.3629
   -0.0151   -0.1007
   -0.5248   -0.5495
   -1.0366   -0.9809
   -1.4863   -1.4010
   -1.8157   -1.7811
   -1.9806   -2.0418
   -1.9578   -2.0777
   -1.7482   -1.8116
   -1.3768   -1.2487
   -0.8892   -0.4962
```

b.





c.

```
xMSE = mean((xHat - x).^2) = 0.6107
yMSE = mean((yHat - y).^2) = 0.0296
```

d.

```
xSNR = 20 * log10(x/(x-xHat)) = 17.9494
ySNR = 20 * log10(y/(y-yHat)) = 12.3148
```

#Problem3.

XDct be the DCT of x, and YDct the DCT of y.

XHatDct be derived from XDct by replacing the last 11 elements of XDct by zeros while keeping the rest of the elements the same YHatDct similarly from YDct.

xHatDct be the inverse DCT of XHatDct, and yHatDct the inverse DCT of YHatDct.

a.

```
table1 = [XDct' XHatDct'];
```

table1 =

21.5000	21.5000
4.5779	4.5779
18.2216	18.2216
0.5019	0.5019
4.4609	4.4609
0.1754	0
1.9048	0
0.0849	0
1.0000	0
0.0469	0
0.5682	0
0.0268	0
0.3170	0
0.0140	0
0.1434	0
0.0044	0

```
table2 = [YDct' YHatDct'];
```

table2 =

-2.0008	-2.0008
4.1666	4.1666
0.9371	0.9371
-1.4285	-1.4285
0.1835	0.1835
-0.3303	0
0.0756	0
-0.1463	0
0.0392	0
-0.0782	0
0.0222	0
-0.0440	0
0.0123	0
-0.0228	0
0.0056	0
-0.0071	0

```
table3 = [x' xHatDct'];
```

table3 =

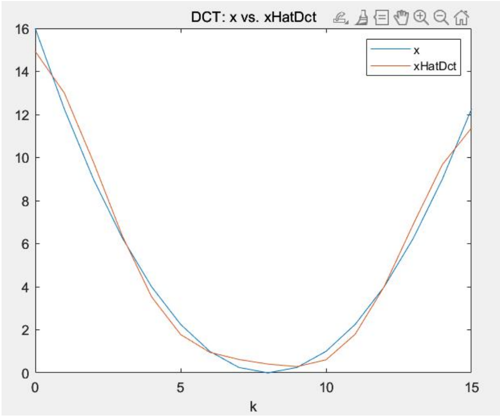
16.0000	14.9312
12.2500	12.9966
9.0000	9.7954
6.2500	6.3422
4.0000	3.5314
2.2500	1.7787
1.0000	0.9546
0.2500	0.6207
0	0.4064
0.2500	0.2893
1.0000	0.6059
2.2500	1.7908
4.0000	4.0072
6.2500	6.9058
9.0000	9.6737
12.2500	11.3701

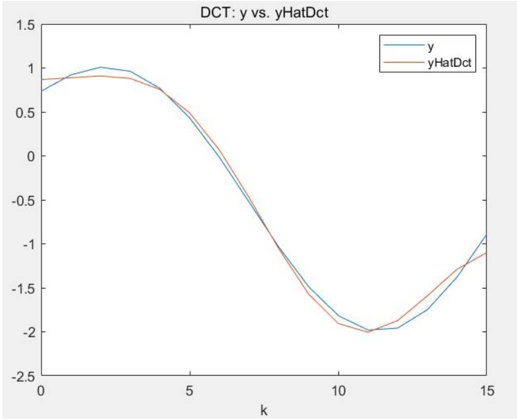
```
table4 = [y' yHatDct'];
```

table4 =

0.7354	0.8674
0.9213	0.8894
1.0086	0.9087
0.9625	0.8813
0.7678	0.7552
0.4320	0.4880
-0.0151	0.0672
-0.5248	-0.4742
-1.0366	-1.0562
-1.4863	-1.5689
-1.8157	-1.9062
-1.9806	-2.0048
-1.9578	-1.8724
-1.7482	-1.5907
-1.3768	-1.2892
-0.8892	-1.0980

b.





c.

`xMSE = mean((xHatDct - x).^2) = 0.3196`

`yMSE = mean((yHatDct - y).^2) = 0.0092`

d.

`xSNR = 20 * log10(abs(x) / abs(x-xHatDct)) = 21.3106`

`ySNR = 20 * log10(abs(y) / abs(y-yHatDct)) = 20.0053`

#Problem4.

X_Hadam and *Y_Hadam* are the Hadamard transforms of x and y.

X_Hadam_abs is the magnitudes of the elements of *X_Hadam*,

Y_Hadam_abs is the same.

X_Hadam_Hat is derived from *X_Hadam* by replacing the 11 smallest-magnitude elements of *X_Hadam* by zeros while keeping the rest of the elements the same.

Y_Hadam_Hat is defined similarly.

smallx_Hadam_hat is the inverse Hadamard transforms of *X_Hadam_Hat*,

smally_Hadam_hat is the same.

a.

`table1 = [X_Hadam X_Hadam_abs X_Hadam_Hat];`

```
table1 =

   21.5000   21.5000   21.5000
    0.5000    0.5000         0
    1.0000    1.0000         0
    1.0000    1.0000         0
    2.0000    2.0000         0
    2.0000    2.0000         0
    4.0000    4.0000         0
         0         0         0
    4.0000    4.0000    4.0000
    4.0000    4.0000    4.0000
    8.0000    8.0000    8.0000
         0         0         0
   16.0000   16.0000   16.0000
         0         0         0
         0         0         0
         0         0         0
```

```
table2 = [Y_Hadam Y_Hadam_abs Y_Hadam_Hat];
```

```
table2 =

   -2.0008    2.0008   -2.0008
    0.1558    0.1558         0
    0.3146    0.3146         0
    0.0392    0.0392         0
    0.6552    0.6552    0.6552
    0.0816    0.0816         0
    0.1649    0.1649         0
   -0.0128    0.0128         0
    4.1447    4.1447    4.1447
    0.1971    0.1971         0
    0.3980    0.3980         0
   -0.2422    0.2422         0
    0.8288    0.8288    0.8288
   -0.5743    0.5743         0
   -1.1919    1.1919   -1.1919
   -0.0162    0.0162         0
```

```
table3 = [x' smallx_Hadam_hat];
```

```
table3 =

   16.0000   13.3750
   12.2500   11.3750
    9.0000    9.3750
    6.2500    7.3750
    4.0000    5.3750
    2.2500    3.3750
    1.0000    1.3750
    0.2500   -0.6250
         0   -2.6250
    0.2500   -0.6250
    1.0000    1.3750
    2.2500    3.3750
    4.0000    5.3750
    6.2500    7.3750
    9.0000    9.3750
   12.2500   11.3750
```

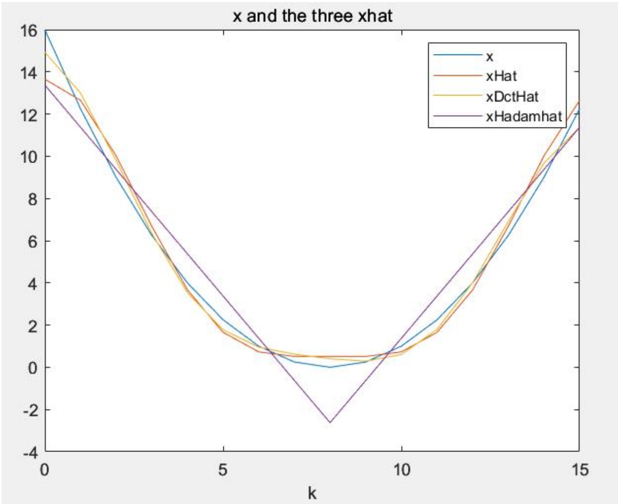
```
table4 = [y' smally_Hadam_hat];
```

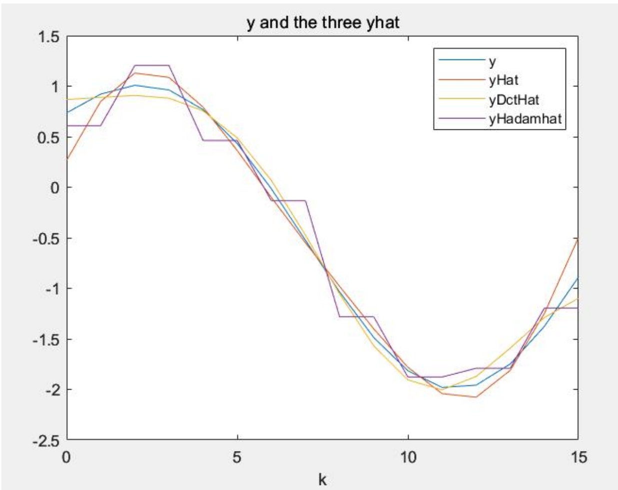
```
table4 =

    0.7354    0.6090
    0.9213    0.6090
    1.0086    1.2050
    0.9625    1.2050
    0.7678    0.4630
    0.4320    0.4630
   -0.0151   -0.1330
   -0.5248   -0.1330
   -1.0366   -1.2818
   -1.4863   -1.2818
   -1.8157   -1.8778
   -1.9806   -1.8778
   -1.9578   -1.7910
   -1.7482   -1.7910
   -1.3768   -1.1950
   -0.8892   -1.1950
```

b.

```
xMSE = mean((smallx_Hadam_hat' - x).^2) = 1.6406
yMSE = mean((smally_Hadam_hat' - x).^2) = 55.4972
```

c.





d.

xMSE2	0.6107
xMSE3	0.3196
xMSE4	1.6406

yMSE2	0.0296
yMSE3	0.0092
yMSE4	0.0465

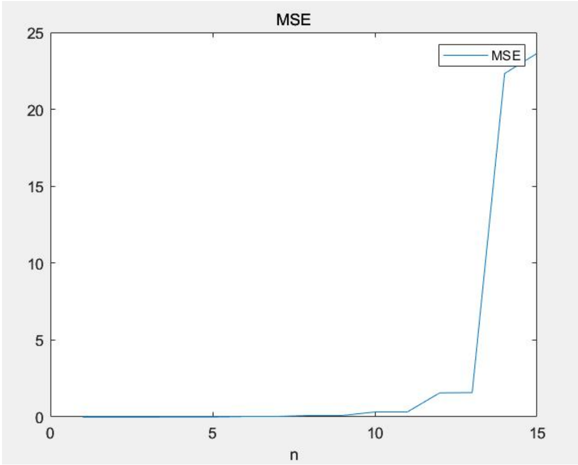
DCT transform gives the best $MSE(x, \hat{x})$.

DCT transform gives the best $MSE(y, \hat{y})$.

#Problem5.

a.

```
>> display(MSE')
    0.0000
    0.0013
    0.0013
    0.0076
    0.0076
    0.0278
    0.0279
    0.0904
    0.0909
    0.3177
    0.3196
    1.5633
    1.5790
   22.3308
   23.6406
```

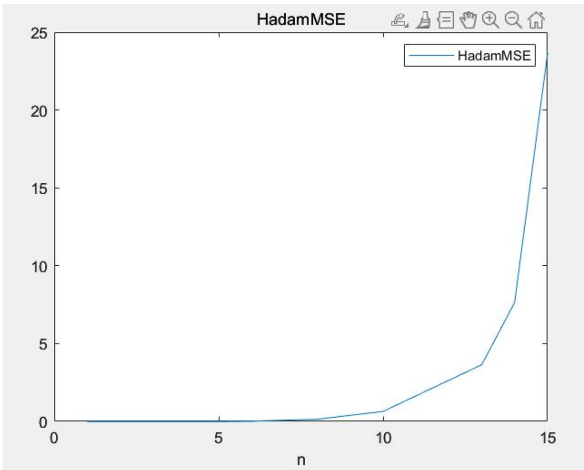


b.

The value of MSE increases as the value of n grows from 1 to 15.

c.

```
>> display(MSE_hadam')
         0
         0
         0
         0
         0
    0.0156
    0.0781
    0.1406
    0.3906
    0.6406
    1.6406
    2.6406
    3.6406
    7.6406
   23.6406
```



The value of MSE increases as the value of n grows from 1 to 15.