FFT_MidTerm_Project

組別：G。B0521229 林威廷、B0521233 陳崇瀚

(a) Please use the Divide-and-Conquer approach to design an efficient algorithm for computing all

$$X[k], \text{ for } 0 \le k \le N-1$$

1. Normal DFT：

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-jk\frac{2\pi}{N}n}$$

2. Divide-and-Conquer

Suppose $N = 2^r$, $W_N = e^{-j\frac{2\pi}{N}}$

$X[k] = x[0] + x[1]W_N^k + x[2]W_N^{2k} + \cdots + x[N-1]W_N^{(N-1)k}$

$= x[0] + x[2]W_N^{2k} + \cdots + x[N-2]W_N^{(N-2)k} + W_N^k\left(x[1] + x[3]W_N^{2k} + \cdots + x[N-1]W_N^{(N-2)k} \right)$

$= X_1[k] + W_N^k X_2[k]$

* $X_1[k]$: $\frac{N}{2}$ point DFT of $x[0], x[2], \ldots, x[N-2]$ ， $X_2[k]$: $\frac{N}{2}$ point DFT of $x[1], x[3], \ldots, x[N-1]$

ex. N = 8,

$X[k] = X_1[k] + W_8^k X_2[k]$

$X_1[k]$: $4\ point\ DFT\ of\ x[0], x[2], x[4], x[6]$ ， $X_2[k]$: $4\ point\ DFT\ of\ x[1], x[3], x[5], x[7]$

$X_1[k] = X_3[k] + W_4^k X_4[k]$ ， k=0,1,2,3

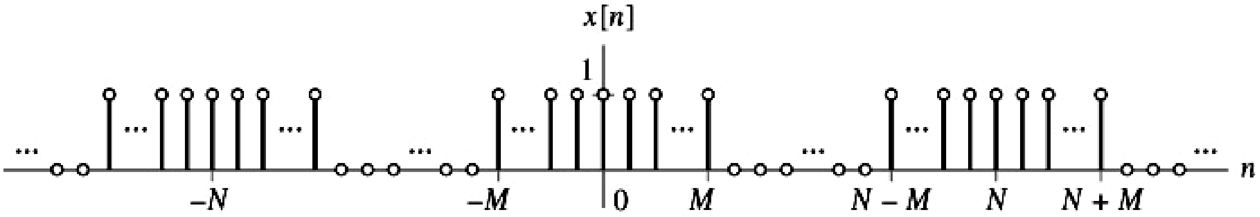
$X_2[k] = X_5[k] + W_4^k X_6[k]$ ， k=0,1,2,3

$X_3[k]$: $2\ point\ DFT\ of\ x[0], x[4]$ ， $X_4[k]$: $2\ point\ DFT\ of\ x[2], x[6]$

$X_5[k]$: $2\ point\ DFT\ of\ x[1], x[5]$ ， $X_6[k]$: $2\ point\ DFT\ of\ x[3], x[7]$

(b) Please implement your algorithm by writing a programming for calculate the DTFS of the periodic signal $x[n]$ with (N, M)= (32768, 10000), where $x[n]$ is given by

$$x[n] = \begin{cases} 1, & mN - M \le n \le mN + M, \\ 0, & mN + M < n < (m+1)N - M, \end{cases}$$ where m is an arbitrary integer.



Divide-and-Conquer part

```
void RecursiveFFT(CArray& x)
{
    //for example N=8
    const size_t N = x.size();
    if (N <= 1) return;

    // divide
    //even=x[0] x[2] x[4] x[6]
    CArray even = x[std::slice(0, N / 2, 2)];
    //even=x[1] x[3] x[5] x[7]
    CArray  odd = x[std::slice(1, N / 2, 2)];

    // conquer
    RecursiveFFT(even);
    RecursiveFFT(odd);


    // combine
    for (size_t k = 0; k < N / 2; ++k)
    {
        Complex t = std::polar(1.0, -2 * PI * k / N) * odd[k];
        x[k] = even[k] + t;
        x[k + N / 2] = even[k] - t;
    }
}
```

Whole source code please see FFT.cpp

Signals test

1. N = 32, M = 5

   x[n] = 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Ans：

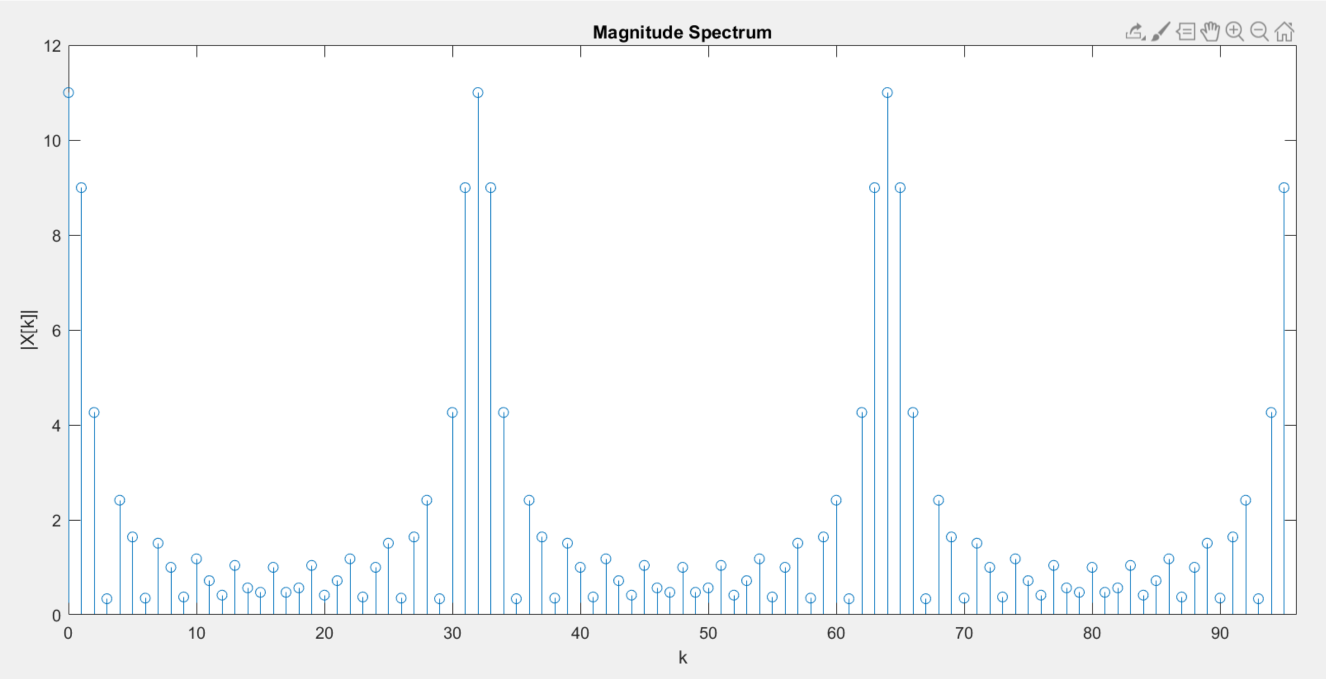
   X[k] =

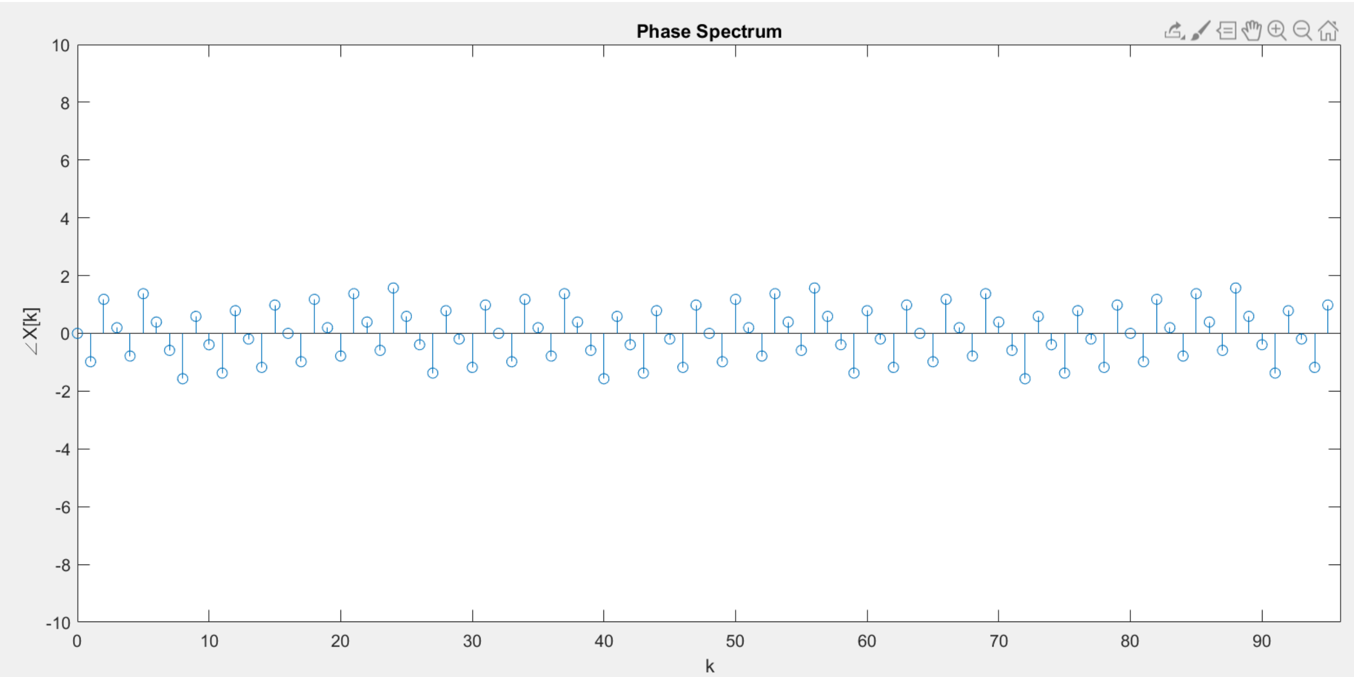
```
------------------After Executing FFT(Divide And Conquer)----------------------
Point      0-----(real,imaginary)parts separated-----is : (11,0)
Point      1-----(real,imaginary)parts separated-----is : (4.99881,-7.48125)
Point      2-----(real,imaginary)parts separated-----is : (-1.63099,-3.93755)
Point      3-----(real,imaginary)parts separated-----is : (0.331171,0.0658739)
Point      4-----(real,imaginary)parts separated-----is : (1.70711,-1.70711)
Point      5-----(real,imaginary)parts separated-----is : (-0.319915,-1.60832)
Point      6-----(real,imaginary)parts separated-----is : (0.324423,0.134381)
Point      7-----(real,imaginary)parts separated-----is : (1.25422,-0.838041)
Point      8-----(real,imaginary)parts separated-----is : (6.12303e-017,-1)
Point      9-----(real,imaginary)parts separated-----is : (0.312238,0.20863)
Point     10-----(real,imaginary)parts separated-----is : (1.08979,-0.451406)
Point     11-----(real,imaginary)parts separated-----is : (0.140335,-0.705509)
Point     12-----(real,imaginary)parts separated-----is : (0.292893,0.292893)
Point     13-----(real,imaginary)parts separated-----is : (1.01998,-0.202887)
Point     14-----(real,imaginary)parts separated-----is : (0.216773,-0.523336)
Point     15-----(real,imaginary)parts separated-----is : (0.263161,0.393849)
Point     16-----(real,imaginary)parts separated-----is : (1,0)
Point     17-----(real,imaginary)parts separated-----is : (0.263161,-0.393849)
Point     18-----(real,imaginary)parts separated-----is : (0.216773,0.523336)
Point     19-----(real,imaginary)parts separated-----is : (1.01998,0.202887)
Point     20-----(real,imaginary)parts separated-----is : (0.292893,-0.292893)
Point     21-----(real,imaginary)parts separated-----is : (0.140335,0.705509)
Point     22-----(real,imaginary)parts separated-----is : (1.08979,0.451406)
Point     23-----(real,imaginary)parts separated-----is : (0.312238,-0.20863)
Point     24-----(real,imaginary)parts separated-----is : (-6.12303e-017,1)
Point     25-----(real,imaginary)parts separated-----is : (1.25422,0.838041)
Point     26-----(real,imaginary)parts separated-----is : (0.324423,-0.134381)
Point     27-----(real,imaginary)parts separated-----is : (-0.319915,1.60832)
Point     28-----(real,imaginary)parts separated-----is : (1.70711,1.70711)
Point     29-----(real,imaginary)parts separated-----is : (0.331171,-0.0658739)
Point     30-----(real,imaginary)parts separated-----is : (-1.63099,3.93755)
Point     31-----(real,imaginary)parts separated-----is : (4.99881,7.48125)
```

*Matlab check * => It matches

```
Command Window
  Columns 1 through 5

  11.0000 + 0.0000i   4.9988 - 7.4813i  -1.6310 - 3.9375i   0.3312 + 0.0659i   1.7071 - 1.7071i

  Columns 6 through 10

  -0.3199 - 1.6083i   0.3244 + 0.1344i   1.2542 - 0.8380i   0.0000 - 1.0000i   0.3122 + 0.2086i

  Columns 11 through 15

   1.0898 - 0.4514i   0.1403 - 0.7055i   0.2929 + 0.2929i   1.0200 - 0.2029i   0.2168 - 0.5233i

  Columns 16 through 20

   0.2632 + 0.3938i   1.0000 + 0.0000i   0.2632 - 0.3938i   0.2168 + 0.5233i   1.0200 + 0.2029i

  Columns 21 through 25

   0.2929 - 0.2929i   0.1403 + 0.7055i   1.0898 + 0.4514i   0.3122 - 0.2086i   0.0000 + 1.0000i

  Columns 26 through 30

   1.2542 + 0.8380i   0.3244 - 0.1344i  -0.3199 + 1.6083i   1.7071 + 1.7071i   0.3312 - 0.0659i

  Columns 31 through 32

fx -1.6310 + 3.9375i   4.9988 + 7.4813i
```

Plot



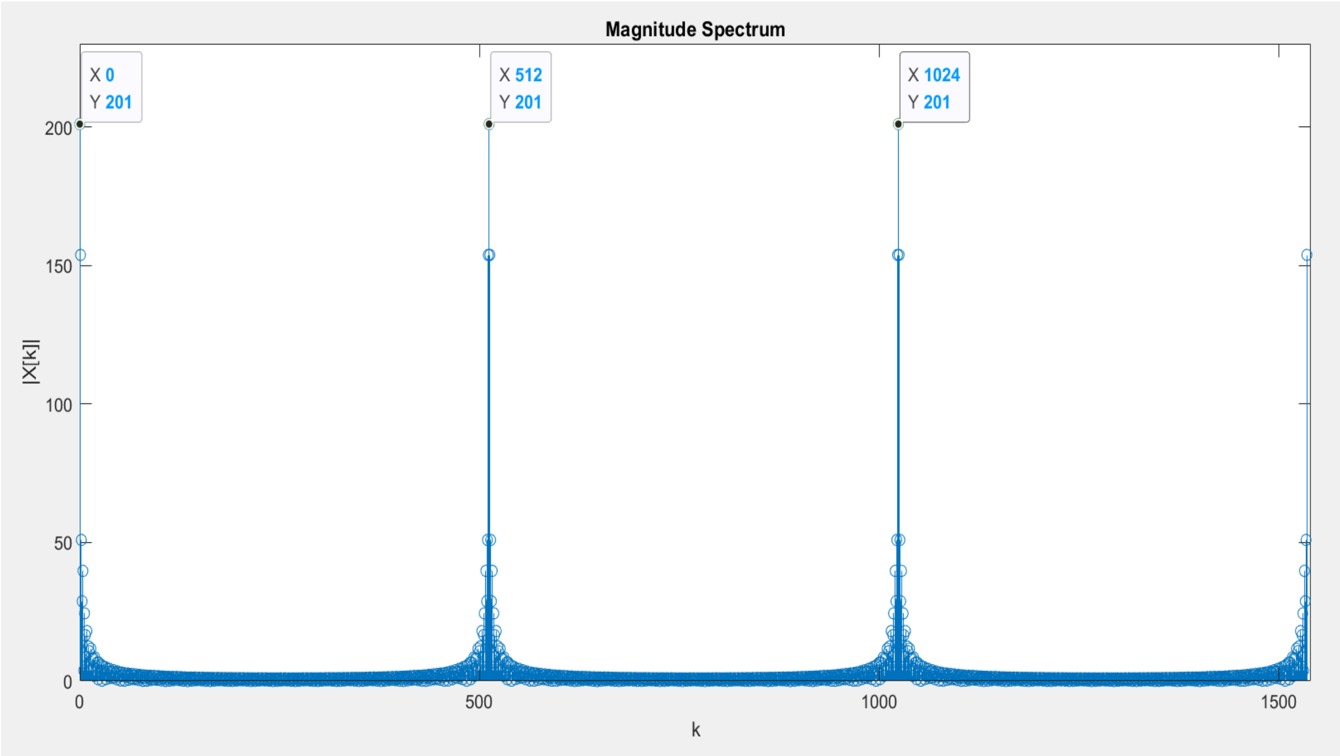


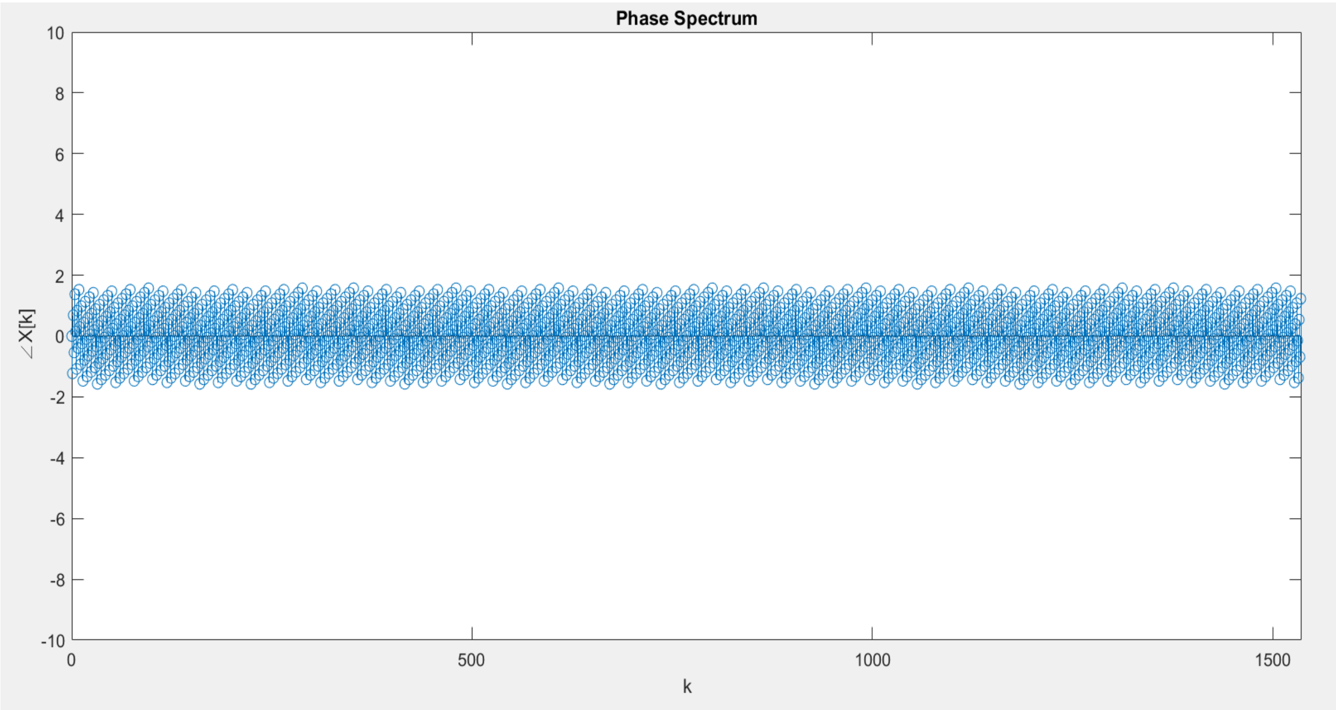
2. N = 512, M = 100

$x[0] \sim x[200] = 1, \quad x[201] \sim x[511] = 0$

Ans：

Plot





3. N = 32768, M = 10000

x[0] ~ x[20000] = 1,   x[20001] ~ x[32767] = 0

Result of the above problem(For first 32 points, you can press y to continue)

(1.) DFT(Straight Forward)

```
------------------After Executing DFT(StraightForward)------------------------
Point      0-----(real,imaginary)parts separated-----is : (20001,0)
Point      1-----(real,imaginary)parts separated-----is : (-3333.04,-9225.89)
Point      2-----(real,imaginary)parts separated-----is : (2564.13,-2130.79)
Point      3-----(real,imaginary)parts separated-----is : (-1517.04,-890.404)
Point      4-----(real,imaginary)parts separated-----is : (469.264,-2520.41)
Point      5-----(real,imaginary)parts separated-----is : (334.226,-54.83)
Point      6-----(real,imaginary)parts separated-----is : (-739.759,-1324.74)
Point      7-----(real,imaginary)parts separated-----is : (738.05,-850.316)
Point      8-----(real,imaginary)parts separated-----is : (-436.918,-168.538)
Point      9-----(real,imaginary)parts separated-----is : (24.8817,-1158.42)
Point     10-----(real,imaginary)parts separated-----is : (316.683,-106.778)
Point     11-----(real,imaginary)parts separated-----is : (-461.555,-580.335)
Point     12-----(real,imaginary)parts separated-----is : (388.469,-630.446)
Point     13-----(real,imaginary)parts separated-----is : (-159.358,-33.2259)
Point     14-----(real,imaginary)parts separated-----is : (-103.733,-730.147)
Point     15-----(real,imaginary)parts separated-----is : (288.674,-153.163)
Point     16-----(real,imaginary)parts separated-----is : (-323.831,-293.503)
Point     17-----(real,imaginary)parts separated-----is : (215.732,-525.377)
Point     18-----(real,imaginary)parts separated-----is : (-23.8601,-1.02546)
Point     19-----(real,imaginary)parts separated-----is : (-156.58,-499.577)
Point     20-----(real,imaginary)parts separated-----is : (251.947,-191.693)
Point     21-----(real,imaginary)parts separated-----is : (-225.709,-145.857)
Point     22-----(real,imaginary)parts separated-----is : (104.026,-450.307)
Point     23-----(real,imaginary)parts separated-----is : (54.7301,-6.57992)
Point     24-----(real,imaginary)parts separated-----is : (-174.334,-346.343)
Point     25-----(real,imaginary)parts separated-----is : (208.762,-220.621)
Point     26-----(real,imaginary)parts separated-----is : (-146.118,-63.6999)
Point     27-----(real,imaginary)parts separated-----is : (24.8242,-384.772)
Point     28-----(real,imaginary)parts separated-----is : (100.569,-29.1645)
Point     29-----(real,imaginary)parts separated-----is : (-170.754,-234.72)
Point     30-----(real,imaginary)parts separated-----is : (161.725,-238.853)
Point     31-----(real,imaginary)parts separated-----is : (-79.2731,-20.1155)
```

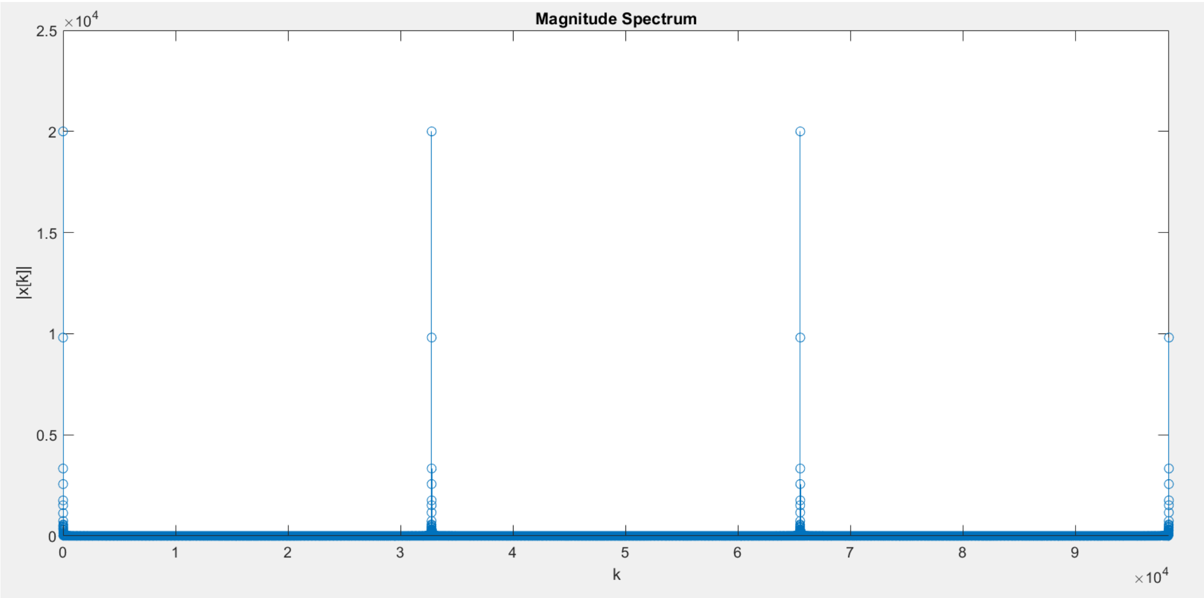
(2.) FFT(Divide and Conquer)

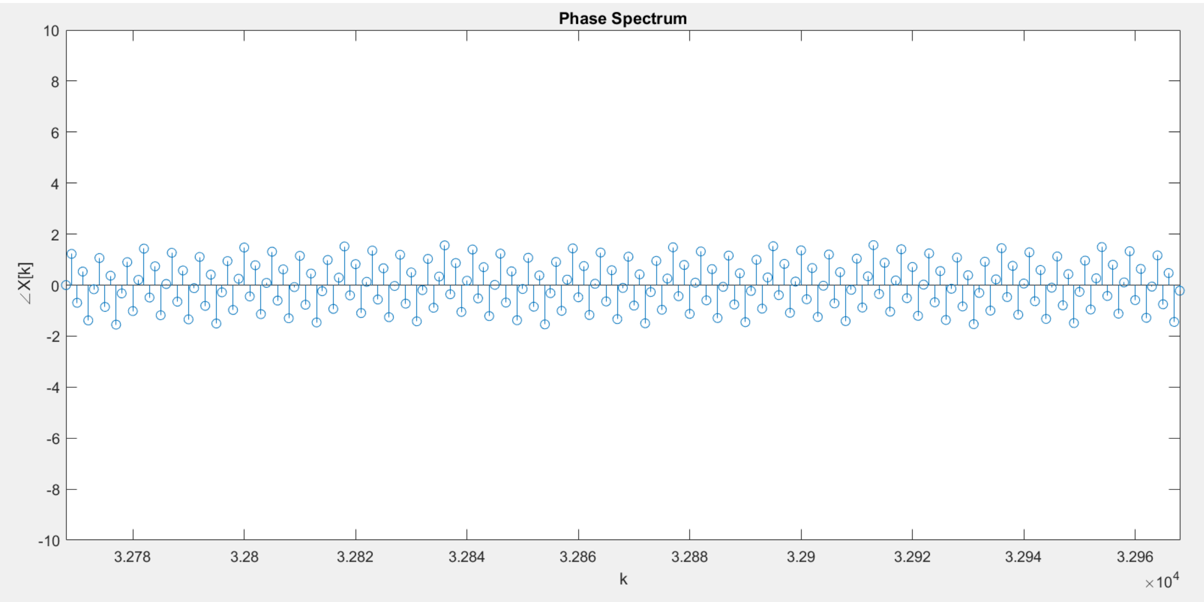
```
------------------After Executing FFT(Divide And Conquer)------------------------
Point      0-----(real,imaginary)parts separated-----is : (20001,0)
Point      1-----(real,imaginary)parts separated-----is : (-3333.04,-9225.89)
Point      2-----(real,imaginary)parts separated-----is : (2564.13,-2130.79)
Point      3-----(real,imaginary)parts separated-----is : (-1517.04,-890.404)
Point      4-----(real,imaginary)parts separated-----is : (469.264,-2520.41)
Point      5-----(real,imaginary)parts separated-----is : (334.226,-54.83)
Point      6-----(real,imaginary)parts separated-----is : (-739.759,-1324.74)
Point      7-----(real,imaginary)parts separated-----is : (738.05,-850.316)
Point      8-----(real,imaginary)parts separated-----is : (-436.918,-168.538)
Point      9-----(real,imaginary)parts separated-----is : (24.8817,-1158.42)
Point     10-----(real,imaginary)parts separated-----is : (316.683,-106.778)
Point     11-----(real,imaginary)parts separated-----is : (-461.555,-580.335)
Point     12-----(real,imaginary)parts separated-----is : (388.469,-630.446)
Point     13-----(real,imaginary)parts separated-----is : (-159.358,-33.2259)
Point     14-----(real,imaginary)parts separated-----is : (-103.733,-730.147)
Point     15-----(real,imaginary)parts separated-----is : (288.674,-153.163)
Point     16-----(real,imaginary)parts separated-----is : (-323.831,-293.503)
Point     17-----(real,imaginary)parts separated-----is : (215.732,-525.377)
Point     18-----(real,imaginary)parts separated-----is : (-23.8601,-1.02546)
Point     19-----(real,imaginary)parts separated-----is : (-156.58,-499.577)
Point     20-----(real,imaginary)parts separated-----is : (251.947,-191.693)
Point     21-----(real,imaginary)parts separated-----is : (-225.709,-145.857)
Point     22-----(real,imaginary)parts separated-----is : (104.026,-450.307)
Point     23-----(real,imaginary)parts separated-----is : (54.7301,-6.57992)
Point     24-----(real,imaginary)parts separated-----is : (-174.334,-346.343)
Point     25-----(real,imaginary)parts separated-----is : (208.762,-220.621)
Point     26-----(real,imaginary)parts separated-----is : (-146.118,-63.6999)
Point     27-----(real,imaginary)parts separated-----is : (24.8242,-384.772)
Point     28-----(real,imaginary)parts separated-----is : (100.569,-29.1645)
Point     29-----(real,imaginary)parts separated-----is : (-170.754,-234.72)
Point     30-----(real,imaginary)parts separated-----is : (161.725,-238.853)
Point     31-----(real,imaginary)parts separated-----is : (-79.2731,-20.1155)
```

(3.)FFT(Bit Reversal)

```
------------------After Executing FFT(BitReversal)------------------------------
Point      0-----(real,imaginary)parts separated-----is : (20001,0)
Point      1-----(real,imaginary)parts separated-----is : (-3333.04,-9225.89)
Point      2-----(real,imaginary)parts separated-----is : (2564.13,-2130.79)
Point      3-----(real,imaginary)parts separated-----is : (-1517.04,-890.404)
Point      4-----(real,imaginary)parts separated-----is : (469.264,-2520.41)
Point      5-----(real,imaginary)parts separated-----is : (334.226,-54.83)
Point      6-----(real,imaginary)parts separated-----is : (-739.759,-1324.74)
Point      7-----(real,imaginary)parts separated-----is : (738.05,-850.316)
Point      8-----(real,imaginary)parts separated-----is : (-436.918,-168.538)
Point      9-----(real,imaginary)parts separated-----is : (24.8817,-1158.42)
Point     10-----(real,imaginary)parts separated-----is : (316.683,-106.778)
Point     11-----(real,imaginary)parts separated-----is : (-461.555,-580.335)
Point     12-----(real,imaginary)parts separated-----is : (388.469,-630.446)
Point     13-----(real,imaginary)parts separated-----is : (-159.358,-33.2259)
Point     14-----(real,imaginary)parts separated-----is : (-103.733,-730.147)
Point     15-----(real,imaginary)parts separated-----is : (288.674,-153.163)
Point     16-----(real,imaginary)parts separated-----is : (-323.831,-293.503)
Point     17-----(real,imaginary)parts separated-----is : (215.732,-525.377)
Point     18-----(real,imaginary)parts separated-----is : (-23.8601,-1.02546)
Point     19-----(real,imaginary)parts separated-----is : (-156.58,-499.577)
Point     20-----(real,imaginary)parts separated-----is : (251.947,-191.693)
Point     21-----(real,imaginary)parts separated-----is : (-225.709,-145.857)
Point     22-----(real,imaginary)parts separated-----is : (104.026,-450.307)
Point     23-----(real,imaginary)parts separated-----is : (54.7301,-6.57992)
Point     24-----(real,imaginary)parts separated-----is : (-174.334,-346.343)
Point     25-----(real,imaginary)parts separated-----is : (208.762,-220.621)
Point     26-----(real,imaginary)parts separated-----is : (-146.118,-63.6999)
Point     27-----(real,imaginary)parts separated-----is : (24.8242,-384.772)
Point     28-----(real,imaginary)parts separated-----is : (100.569,-29.1645)
Point     29-----(real,imaginary)parts separated-----is : (-170.754,-234.72)
Point     30-----(real,imaginary)parts separated-----is : (161.725,-238.853)
Point     31-----(real,imaginary)parts separated-----is : (-79.2731,-20.1155)
```

Plot





(c) Compare the computation time of your algorithm with that of the direct computation scheme

Execution Time

```
Please wait for about one minute,three procedures are running
            Time taken by procedure  Normal-DFT  is :64.126seconds
            Time taken by procedure RecursiveFFT is :0.022seconds
            Time taken by procedure IterativeFFT is :0.004seconds
```

(1.) Direct computation

67.531seconds

(2.) Divide-and-Conquer

0.025seconds

(3.) Bit-Reversal

0.004seconds

Time Complexity Analysis

By using the method known as the Fast Fourier Transfrom(FFT),we can compute DFT in time θ(NlgN), as opposed to the $\theta(N^2)$ time of the straightforward method.

(1.) Direct Computation $\in \theta(N^2)$
(2.) Divide-and-Conquer $\in \theta(NlgN)$

As discussion in (a), We have T(n)=2T(n/2)+ θ(N),which leads to $T(n) \in \theta(NlgN)$

(3.) Bit reversal $\in \theta(NlgN)$

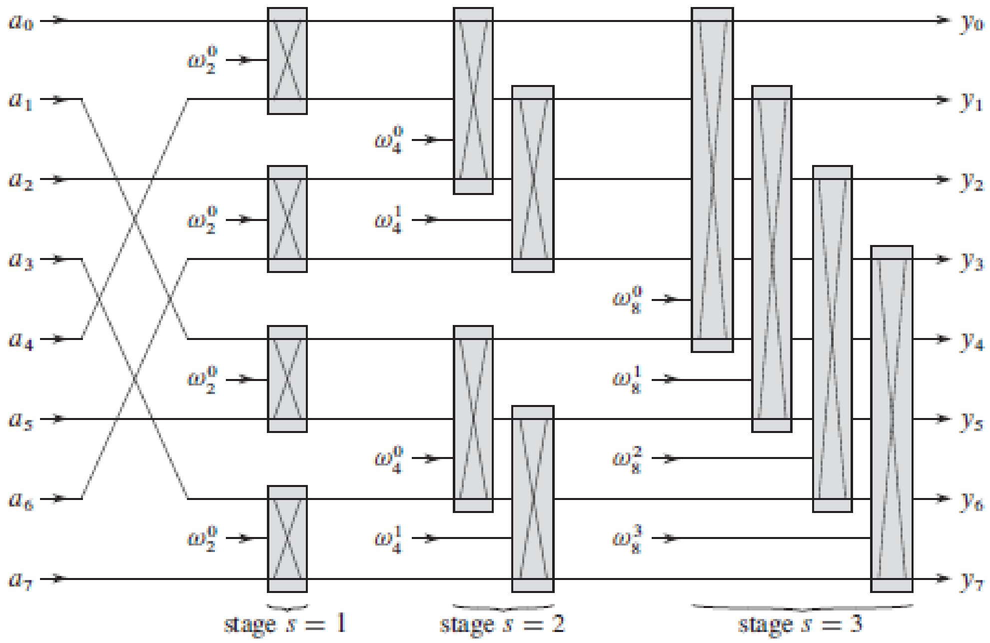


Figure 30.5 A circuit that computes the FFT in parallel, here shown on $n = 8$ inputs. Each butterfly operation takes as input the values on two wires, along with a twiddle factor, and it produces as outputs the values on two wires. The stages of butterflies are labeled to correspond to iterations of the outermost loop of the ITERATIVE-FFT procedure. Only the top and bottom wires passing through a butterfly interact with it; wires that pass through the middle of a butterfly do not affect that butterfly, nor are their values changed by that butterfly. For example, the top butterfly in stage 2 has nothing to do with wire 1 (the wire whose output is labeled y_1); its inputs and outputs are only on wires 0 and 2 (labeled y_0 and y_2, respectively). This circuit has depth $\Theta(\lg n)$ and performs $\Theta(n \lg n)$ butterfly operations altogether.

which also leads to $T(n) \in \theta(NlgN)$

Reference:
https://www.amazon.com/Introduction-Algorithms-Press-Thomas-Cormen-ebook/dp/B007CNRCAO