FFT 1 and 2

September 8, 2021 10:15 PM

$$W = [1, \omega, \omega^2, \dots, \omega^{n/2-1}, 1, \omega^2, \omega^4, \dots, \omega^{n/2-2}, 1, \omega^4, \omega^8, \dots, \omega^{n/2-4}, \dots, 1, 0]$$

```
void FFT1( int *A, int n,
           int *W, int p, int *T )
{
  int i,n2,t;
  if( n==1 ) return;
  n2 = n/2;
  for( i=0; i<n2; i++ ) {
     T[   i] = A[2*i];
     T[n2+i] = A[2*i+1];
  }
  FFT1( T,    n2, W+n2, p, A );
  FFT1( T+n2, n2, W+n2, p, A+n2 );
  for( i=0; i<n2; i++ ) {
     t = mulmod(W[i],T[n2+i],p);
     A[   i] = addmod(T[i],t,p);
     A[n2+i] = submod(T[i],t,p);
  }
  return;
}
```

```
void FFT2( int *A, int n,
           int *W, int p, int *T )
{
  int i,n2,t;
  if( n==1 ) return;
  n2 = n/2;
  for( i=0; i<n2; i++ ) {
     T[i] = addmod(A[i],A[n2+i],p);
     t = submod(A[i],A[n2+i],p);
     T[n2+i] = mulmod(t,W[i],p);
  }
  FFT2( T,    n2, W+n2, p, A );
  FFT2( T+n2, n2, W+n2, p, A+n2 );
  for( i=0; i<n2; i++ ) {
     A[ 2*i] = T[i];
     A[2*i+1] = T[n2+i];
  }
  return;
}
```


FFI

$A = [a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7]$

$T = [a_0, a_2, a_4, a_6, a_1, a_3, a_5, a_7]$

$A = [a_0\, a_4\, a_2\, a_6\, a_1\, a_5\, a_3\, a_7]$

$T = [a_0\, a_4\, a_2\, a_6\, a_1\, a_5\, a_3\, a_7]$

$\pi = (0\ 4\ 2\ 6\ 1\ 5\ 3\ 7)$

$A = [a_0\, a_1\, a_2\, a_3\, a_4\, a_5\, a_6\, a_7]$

The two permutations

$A = [a_0\ a_1\ \dots\ a_7]$

$A = [\boxed{b_0 \mid b_1} \boxed{b_2 \mid b_3}\ b_4\ b_5\ b_6\ b_7]$

$A = [\boxed{b_0\ b_1\ b_2\ b_3} \boxed{b_4\ b_5\ b_6\ b_7}]$

$A = [b_0\ b_2\ b_1\ b_3\ b_4\ b_6\ b_5\ b_7]$

$T = [b_0\ b_4\ b_2\ b_6\ b_1\ b_5\ b_3\ b_7]$

0 2 4 6

$\pi = (0\ 4\ 2\ 6\ 1\ 5\ 3\ 7).$

In-place FFT routines with permutation removed.

```
void FFT1( int *A, int n,
           int *W, int p )
{
  int i,n2,t;
  if( n==1 ) return;
  n2 = n/2;
  FFT1( A,    n2, W+n2, p );
  FFT1( A+n2, n2, W+n2, p );
  for( i=0; i<n2; i++ ) {
     s = A[i];
     t = mulmod(W[i],A[n2+i],p);
     A[   i] = addmod(s,t,p);
     A[n2+i] = submod(t,t,p);
  }
  return;
}
```

```
void FFT2( int *A, int n,
           int *W, int p )
{
  int i,n2,t;
  if( n==1 ) return;
  n2 = n/2;
  for( i=0; i<n2; i++ ) {
     s = addmod(A[i],A[n2+i],p);
     t = submod(A[i],A[n2+i],p);
     A[i] = s;
     A[n2+i] = mulmod(t,W[i],p);
  }
  FFT2( A,    n2 W+n2, p );
  FFT2( A+n2, n2, W+n2, p );
  return;
}
```

$$W = [1, \omega, \omega^2, ..., \omega^{n/2-1}, 1, \omega^2, \omega^4, ..., \omega^{n/2-2}, 1, \omega^4, \omega^8, ..., \omega^{n/2-4}, ..., 1, 0]$$