3b_test1

April 1, 2019

```
In [1]: from cmath import exp, pi

        def fft(x):
            N = len(x)
            if N <= 1: return x
            even_part = fft(x[0::2])
            odd_part =  fft(x[1::2])
            T= [exp(-2j*pi*k/N)*odd_part[k] for k in range(N//2)]
            return [even_part[k] + T[k] for k in range(N//2)] + \
                   [even_part[k] - T[k] for k in range(N//2)]
```

```
In [2]: from numpy import array
```

```
In [3]: import numpy as np
```

```
In [4]: a = array([0, -4j, 0, 0, 0, 0, 0, 4j])
```

0.1 From here we are Implimenting IFFT

IFFT(X)=1/Nconj(FFT(conj(X)))

```
In [5]: N = len(a)
        ifft_output = (1/N)*np.conj(fft(np.conj(a)))
```

0.2 Lets see the output result of IFFT on a using our IFFT Function

```
In [6]: ifft_output
```

```
Out[6]: array([ 0.        +0.00000000e+00j,  0.70710678+0.00000000e+00j,
                1.        -6.12323400e-17j,  0.70710678-1.11022302e-16j,
                0.        +0.00000000e+00j, -0.70710678+0.00000000e+00j,
               -1.        +6.12323400e-17j, -0.70710678+1.11022302e-16j])
```

0.3 Lets see the output result of IFFT on a using Numpy's IFFT Function

```
In [7]: np.fft.ifft(a)
```

```
Out[7]: array([ 0.        +0.j,  0.70710678+0.j,  1.        +0.j,  0.70710678+0.j,
                0.        +0.j, -0.70710678+0.j, -1.        +0.j, -0.70710678+0.j])
```

0.4 Here we are comparing our result with Numpy's ifft

```
In [8]: np.allclose(ifft_output, np.fft.ifft(a))
```

```
Out[8]: True
```

0.5 Yes, As we can see that two results are element-wise equal within a tolerance.

0.6 Now lets see what is the difference between our IFFT output and Numpy's IFFT

```
In [9]: ifft_output - np.fft.ifft(a)
```

```
Out[9]: array([ 0.00000000e+00+0.00000000e+00j,  0.00000000e+00+0.00000000e+00j,
                0.00000000e+00-6.12323400e-17j, -1.11022302e-16-1.11022302e-16j,
                0.00000000e+00+0.00000000e+00j,  0.00000000e+00+0.00000000e+00j,
                0.00000000e+00+6.12323400e-17j,  1.11022302e-16+1.11022302e-16j])
```

0.7 Hurray!!! , The both outputs are equal (almost, difference is in around 10ˆ-17)