# FFT_test1

April 1, 2019

```
In [7]: from cmath import exp, pi

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

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

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

```
In [4]: a = array(np.random.random(1024))
```

```
In [5]: a
```

```
Out[5]: array([0.48228037, 0.7726821 , 0.19520506, ..., 0.95585951, 0.71307254,
               0.73051064])
```

```
In [9]: array(fft_fn(a))
```

```
Out[9]: array([504.80157023+0.j        ,   1.16209287+1.87240243j,
                 6.63018469-0.4463443j , ...,   1.33626387+1.41730398j,
                 6.63018469+0.4463443j ,   1.16209287-1.87240243j])
```

```
In [10]: np.fft.fft(a)
```

```
Out[10]: array([504.80157023+0.j        ,   1.16209287+1.87240243j,
                  6.63018469-0.4463443j , ...,   1.33626387+1.41730398j,
                  6.63018469+0.4463443j ,   1.16209287-1.87240243j])
```

```
In [11]: np.allclose(array(fft_fn(a)), np.fft.fft(a))
```

```
Out[11]: True
```