B0729036 林紹翰

3.

(a) convolution Theorem 指,函數摺積的傅立葉轉換是函數傅立葉轉換的乘積，即一個域中摺積對應於另一個域中的乘積。

(b) $L[f(t)*g(t)] = \int_0^{\infty}\int_0^{t} f(\tau)\, g(t-\tau)\, d\tau\, e^{-st}\, dt = \int_0^{\infty} f(\tau) \int_{\tau}^{\infty} g(t-\tau)\, e^{-st}\, dt\, d\tau$

$= \int_0^{\infty} g(x)\, e^{-sx}\, dx \times \int_0^{\infty} f(\tau)\, e^{-s\tau}\, d\tau = G(s) \cdot F(s)$

1.

```
import math
def iexp(n): return complex(math.cos(n), math.sin(n))
def is_pow2(n): return False if n==0 else (n==1 or is_pow2(n>>1))
def dft(xs): "naive dft" n=len(xs) return [sum((xs[k]*iexp(-2*math.pi*i*k/n)
for k in range(n)))
for i in range(n)]
def dftinv(xs): "naive dft" n=len(xs)
return [sum((xs[k]*iexp(2*math.pi*i*k/n) for k in range(n)))/n
for i in range(n)]  if _name_ == "_main_"
```

(a) wave1 = [1,0,0,0,0,0,0,0] dfreq1 = dft(wave1) print(dfreq1)

$x_1[k]$ = [(1+0j), (1+0j), (1+0j), (1+0j), (1+0j), (1+0j), (1+0j), (1+0j)]

(b) wave2 = [1,1,1,1,1,1,1,1] dfreq2 = dft(wave2) print(dfreq2)

$x_2[k]$ = [(8+0j), (-5.511e-16+2.22e-16j), (-4.28e-16-4.4408e-16j), (-2.22e-16+8.88e-16j), (-4.89e-16j), (-2.109e-15-1.22e-15j), (-2.932e-15-6.661e-16j), (3.552e-15+1.11-15j)]

(c) wave3 = [1,-1,1,-1,1,-1,1,-1] print(dfreq3)

$x_3[k]$ = [0j, (1.11e-16-1, 1.11e-16j), (9.55e-17-1.11-16j), (8.881e-16-1.55e-15j) (8+3.429-15j), (-2.6645e-15+15+1.11-16j), (2.93e-15-6.66e-16j) (-5.2108e-15-2.66453e-15j)]

(d) wave 4 = [3,0,2,0,2,0,2,0] dfreq4 = dft(wave 4) print(dfreq4)

$X_4[k]$ = [(9+0j, (0.99 − 4.44−16j), (1−4.898−16j), (1−6.613e−16j),
(9+2.93e−15j), (0.999 − 1.33e−15j), (1−1.4695e−15j),
(0.99 − 1.77e−15j)]

2.

(a) f(n) = f(0) g(2) + f(1) g(1) + f(2) g(0)

(b) x = np. array ([1,1,1,1])
w = np. array ([1,1])
y = np. convolve
output = array ([1,2,2,2,1])

(c) x = fp. fft (x)
output = arry ([4, −0.j, 0+0.j, 0−0.j, 0, −0.j])
w = fp fft(w)
output = array ([2, −0.j, 0, −6−j])
y = fp. fft (y)
output = array ((f, −0.j, −1.309 − 0.951j, −0.191 − 0.58j, −0.191 + 0.588j,
−1.3.9 + 0.95))