DSP last HW 蔡侑霖 311512038

假設 $N=2^a$, $M=\frac{N}{2}$

給定 N-length sequence : $v[n] \xrightarrow{N\ \text{point DFT}} V[k]$

傳統DFT算法在電腦中實現可以視為以下矩陣乘法

$$\underbrace{\begin{bmatrix} V[0] \\ V[1] \\ \vdots \\ V[N-1] \end{bmatrix}}_{N\times 1} = \underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N & \cdots & W_N^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & \cdots & W_N^{(N-1)^2} \end{bmatrix}}_{N\times N} \underbrace{\begin{bmatrix} v[0] \\ v[1] \\ \vdots \\ v[N-1] \end{bmatrix}}_{N\times 1} = W\vec{v}, \quad \text{where } W_N = e^{-j\frac{2\pi}{N}}$$

矩陣相乘時,所需的計算量為

(i) $W[i][j] \times v[j]$, $i,j=0,1,\ldots,N-1 \Rightarrow N^2$次乘法

(ii) $V[i]=\sum_{j=0}^{N-1} W[i][j]\,v[j]$, $i=0,1,\ldots,N-1$, 使用 $N-1$ 次加法

將N項 $W[i][j]\,v[j]$ 相加,並執行N次 $\Rightarrow N(N-1)$ 次加法

計算量記作:

Multiply : N^2 次 (記作 N^2 MUL)

Add : $N(N-1)$ 次 (記作 $N(N-1)$ ADD)

$\Rightarrow$ Time Complexity : $O(n^2)$

使用Ch5(3)中1~13頁的Algorithm將1個N point sequence拆成2個 $\frac{N}{2}$ point sequence ($g[n]$, $h[n]$)
可將N point DFT轉換成2個 $\frac{N}{2}$ point DFT, 並且可藉由關係式: $V[k]=G[\langle k\rangle_{\frac{N}{2}}]+W_N^k H[\langle k\rangle_{\frac{N}{2}}]$
得到 $V[k]$, 此時計算量的變化由以下表示:
(DFT_N 表示為 N point DFT)

$$\mathrm{DFT}_N \xrightarrow{\text{Algorithm}} 2\mathrm{DFT}_{\frac{N}{2}} + \underline{N\ \mathrm{MUL} + N\ \mathrm{ADD}}$$

對於每個 k：$V[k] = G[\langle k\rangle_{\frac{N}{2}}] + W_N^k H[\langle k\rangle_{\frac{N}{2}}]$：$1\ \mathrm{MUL} + 1\ \mathrm{ADD}$
而 $0 \le k \le N-1$，$V[k] = G[\langle k\rangle_{\frac{N}{2}}] + W_N^k H[\langle k\rangle_{\frac{N}{2}}]$ 的計算量為 $N\ \mathrm{MUL} + N\ \mathrm{ADD}$

以此類推，可得

$$\mathrm{DFT}_{N/2} \xrightarrow{\text{Algorithm}} 2\mathrm{DFT}_{N/4} + \frac{N}{2}\ \mathrm{MUL} + \frac{N}{2}\ \mathrm{ADD}$$
$$\mathrm{DFT}_{N/4} \xrightarrow{\text{Algorithm}} 2\mathrm{DFT}_{N/8} + \frac{N}{4}\ \mathrm{MUL} + \frac{N}{4}\ \mathrm{ADD}$$
$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$
$$\mathrm{DFT}_4 \xrightarrow{\text{Algorithm}} 2\mathrm{DFT}_2 + 4\ \mathrm{MUL} + 4\ \mathrm{ADD}$$

故 $\mathrm{DFT}_N \longrightarrow 2\mathrm{DFT}_{\frac{N}{2}} + N\ \mathrm{MUL} + N\ \mathrm{ADD}$

$$\longrightarrow 4\mathrm{DFT}_{\frac{N}{4}} + 2N\ \mathrm{MUL} + 2N\ \mathrm{ADD}$$
$$\longrightarrow 8\mathrm{DFT}_{\frac{N}{8}} + 3N\ \mathrm{MUL} + 3N\ \mathrm{ADD}$$
$$\vdots \qquad\qquad \vdots$$
$$\longrightarrow \frac{N}{2}\mathrm{DFT}_2 + \log_2\left(\frac{N}{2}\right)N\ \mathrm{MUL} + \log_2\left(\frac{N}{2}\right)N\ \mathrm{ADD}$$

討論 DFT_2 的計算量：令 $x[n] = x[0]\ x[1]$

$$\xrightarrow[\text{DFT}]{\text{2 point}} X[k] = x[0] + x[1]e^{-j\pi k}$$
$$\Rightarrow X[0] = x[0] + x[1], \quad X[1] = x[0] - x[1]$$
$\Rightarrow \mathrm{DFT}_2$ 的計算量為：$2\ \mathrm{ADD}$

因此 $\mathrm{DFT}_N \longrightarrow \frac{N}{2}\mathrm{DFT}_2 + \log_2\left(\frac{N}{2}\right)N\ \mathrm{MUL} + \log_2\left(\frac{N}{2}\right)N\ \mathrm{ADD}$

$$\sim (\log_2 N - 1)N\ \mathrm{MUL} + (\log_2 N)N\ \mathrm{ADD}$$
$$\sim \text{Time Complexity } O(n\log_2 n)$$

(b) 由以上可知，若重覆使用此Algorithm，將 N-length sequence 切割成 $\frac{N}{2}$個 2-length sequence，即可使用數個 2 point DFT 去計算出 N point DFT，稱為 FFT algorithm

(c) 由以上可知，使用 FFT algorithm，能使原本DFT為 $O(n^2)$ 的 Time Complexity 降低至 Time Complexity $= O(n\log_2 n)$，能大幅減少計算時間