Question 7

DFT of N-numbers can be written as

$$\tilde{f}_q = \sum_{K=0}^{N-1} f(K)\, W^{qK} \quad \text{———} \quad (1)$$

where $W = e^{-\frac{i2\pi}{N}}$

Scaling is ignored as it would not affect our calculation of time complexity.

lets assume $N = r_1 r_2$

then

$$q = q_1 r_1 + q_0 \;; \quad q_0 = 0, 1, \ldots, r_1 - 1$$
$$q_1 = 0, 1, \ldots r_2 - 1$$

$$K = K_1 r_2 + K_0 \;; \quad K_0 = 0, 1, \ldots, r_2 - 1$$
$$K_1 = 0, 1, \ldots, r_1 - 1$$

In terms of these indices; eq^n (1) can be written as

$$\tilde{f}(q_0, q_1) = \sum_{K_0} \sum_{K_1} f(K_1, K_0)\, W^{q(K_1 r_2 + K_0)}$$

$$W^{q(K_1 r_2 + K_0)} = W^{K_1 q_1 r_1 r_2} \cdot W^{q_0 K_1 r_2}\, W^{q_1 r_1 K_0}\, W^{q_0 K_0}$$
$$= W^{q_0 K_1 r_2}\, W^{q_1 r_1 K_0}\, W^{q_0 K_0}$$

lets define

$$f_1 = \sum_{K_1} f(K_1, K_0)\, W^{q_0 K_1 r_2}$$

then

$$\tilde{f}(q_0, q_1) = \sum_{K_0} f_1(q_0, K_0)\, W^{(q_1 r_1 + q_0) K_0}$$

There are N ~~operations in~~ elements in $f_1(q_0, K_0)$ each requiring r_1 operations to obtain f_1.

Similarly it takes Nr_2 operations to get $\mathcal{F}(q_0, q_1)$

$\therefore$ this two step algorithm requires $T = N(r_1 + r_2)$ — (2) operations.

For m-step algorithm this generalises to $T = N(r_1 + r_2 + \dots + r_m)$, where $N = r_1 + r_2 + \dots + r_m$. — (3)

Lemma 1

if $r_j = s_j t_j$ then $s_j + t_j < r_j$ where $s_j > 1$, $t_j > 1$, except at $s_j = 2$ $t_j = 2$.

So If N is highly composite then T can be minimised further giving good reduction in time complexity.

Also due to equality in lemma 1 there is no loss if factors of 2 are combined.

$$\text{if } r_j = r \quad \forall j$$

$$\text{then } T(r) = rN\log_r N$$

$$\text{if } r = 2$$

$$T = 2N\log_2(N) \simeq O(N\log N)$$