Manuel Leda

Exercise: Let $k \in \mathbb{N}$ and $n = 2^k$. Let $f = \sum_{0 \le i < n} f_i x^i \in R[x]$, so $\deg f < n$ and $w, w^2, \dots, w^{n-1}$ the powers of a primitive root of unity $w \in R$.
Determine $S(n)$ and $T(n)$ where $S(n)$ is the # of additions in R and $T(n)$ # of multiplications in R of the FFT algorithm for f with $\deg f < n$.

Step 1: if $n = 1$ then return (f_0) — 0 additions and 0 multiplications.

Step 2: $r_0 \leftarrow \sum_{0 \le j < n/2} (f_j + f_{j+n/2}) x^j$ $\qquad$ $r_1^* \leftarrow \sum_{0 \le j < n/2} (f_j - f_{j+n/2}) w^j x^j$

$n/2$ additions
0 multiplications (just shift j)

$n/2$ additions
$n/2$ multiplications

$\Rightarrow$ n additions
$n/2$ multiplications

Step 3: Call the algorithm recursively to evaluate r_0 and r_1^* at the powers of w^2

$\Rightarrow$ Recursion $\quad 2S(n/2) \quad 2T(n/2)$

- $S(1)$ and $T(1)$ $\Rightarrow$ 0 add and 0 mult since return in step 1.
- $S(n)$ and $T(n)$: $\quad S(n) = 2S(n/2) + n$
  $\quad T(n) = 2T(n/2) + n/2$

Therefore,

- $S(n) = 2(S(n/2)) + n = 2\left(2S(n/4) + \frac{n}{2}\right) + n = 2^2 S(n/2^2) + 2n \dots$
  After k iterations can be proven by induction that
  $S(n) = 2^k S(n/k) + kn = 2^k S(1) + kn$ ~~...~~ $= n\log_2 n$
- $T(n) = 2T(n/2) + n/2 = \dots$
  $2^k T(n/2^k) + k\,n/2 = \frac{1}{2} n \log_2 n$

$\Rightarrow n\log_2 n + \frac{1}{2} n\log_2 n = \frac{3}{2} n \log_2 n.$ ☐