

7) FFT algorithm works by breaking the array into two arrays (even and odd) and finding the respective FFT of each of the $N/2$ arrays and this is repeated till we reach only one element in FFT at which the number is itself the FFT

we can write

$\therefore \text{FFT}(N) = \text{FFT}(N/2) + \text{factor} \times (\text{FFT}(N/2))$
which involve $N/2$ additions $N/2$ multiplications
and 2 FFT of $N/2$ arrays

number of steps ~~that~~ in FFT $T(N)$

$$\therefore T(N) = 2T(N/2) + N$$

and $T(N/2)$ can be expanded as $2T(N/4) + N/2$

So we have $\log_2(N)$ such levels with $O(N)$ operations in each level \therefore

The complexity of FFT is $O(N \log_2(N))$

Reference: The Computational Complexity of FFT, Mathias Lohme
2017