Quick Derivation of the Fast Fourier Transform (FFT)

The Discrete Fourier Transform (DFT) is defined as

$$\begin{aligned} X[m] &= FFT(x,m) \\ &= \sum_{n=0}^{2N-1} x[n]e^{-j(2\pi/2N)mn} \\ &= \sum_{n=0}^{2N-1} x[n]e^{-\frac{j2\pi mn}{2N}} \end{aligned}$$

where x[n] is an array of complex numbers of length 2N. If we divide this into even [0,2,4,6,8,…] and odd [1,3,5,7,9,…] indices, we get

$$\begin{aligned} X[m] &= FFT(x,m) \\ &= \sum_{n=0,even}^{2N-1} x[n]e^{-\frac{j2\pi mn}{2N}} + \sum_{n=0,odd}^{2N-1} x[n]e^{-\frac{j2\pi mn}{2N}} \\ &= \sum_{n=0}^{N-1} x[2n]e^{-\frac{j2\pi m(2n)}{2N}} + \sum_{n=0}^{N-1} x[2n+1]e^{-\frac{j2\pi m(2n+1)}{2N}} \\ &= \sum_{n=0}^{N-1} xeven[n]e^{-\frac{j2\pi m(2n)}{2N}} + \sum_{n=0}^{N-1} xodd[n]e^{-\frac{j2\pi m(2n+1)}{2N}} \\ &= \sum_{n=0}^{N-1} xeven[n]e^{-\frac{j2\pi m(2n)}{2N}} + e^{-\frac{j2\pi n}{2N}} \sum_{n=0}^{N-1} xodd[n]e^{-\frac{j2\pi m(2n)}{2N}} \\ &= \sum_{n=0}^{N-1} xeven[n]e^{-\frac{j2\pi m(n)}{N}} + e^{-\frac{j2\pi n}{2N}} \sum_{n=0}^{N-1} xodd[n]e^{-\frac{j2\pi m(n)}{N}} \\ &= FFT(xeven,m) + e^{-\frac{j2\pi n}{2N}} * FFT(xodd,m) \end{aligned}$$

This is a recursive definition, and can be applied to 2^N data points, in N steps, to reduce the problem to finding the FFT of two data points. This results in a calculation that uses $O(N \log N)$ operations rather than $O(N^2)$ calculations.

There are, of course, implementation details that are fairly detailed. A good web site to look at is http://www.fftw.org/ (Fastest Fourier Transform in the West)