# Solutions for: Fast Fourier Transform Tutorial

1. How can the algorithms described here be modified to compute the inverse DFT? *Write out the pseudo code for the modified algorithms*.

2. What is the run time complexity (asymptotic upper bound) for a recursive radix-3 FFT that divides the data into three portions of equal number of terms ($3\tau$, $3\tau + 1$ and $3\tau + 2$)? *Assume $N$ is divisable by 3*. Give all working and use the substitution method if and where necessary.

3. FFTs are often applied to multidimensional data, such as photographs (2D) and volumetric medical imaging data (3D). The equation for a 2D FFT is:

$$X(f_1, f_2) = \sum_{t_2=0}^{N_2} \sum_{t_1=0}^{N_1} \exp\left(2\pi f_2 t_2 / N_2\right) \exp\left(2\pi f_1 t_1 / N_1\right) x(t_1, t_2).$$

   Design an algorithm to compute this 2D FFT as efficiently as possible. What is the computational complexity of your algorithm assuming $N = N_1 = N_2$ and $N$ is divisable by 2? Give all working and use the substitution method if and where necessary.