COMP261 Assignment 3 report

Xiaobin Zhuang
ID: 300519184

Stage 1:

- Successfully implement the DFT and IDFT according to the Fourier transform and Inverse Fourier transform equation. Both work on small files.
- When doing IDFT, converting the obtained complex number list to double list to restore the waveform. Can display the restored waveform by clicking the GUI button.
- Simply adding 3 test cases into waveform list, which given by lecturer. Apply DFT to these 3 cases separately, can get 8 complex numbers of each case. Then apply the IDFT to these outputs, can get 8 double numbers, which is almost same with the number of the test cases. (Rounding errors)

Stage 2:

- Successfully implement the recursive FFT, which refer to [1].
- To judge whether N is the power of two or not, by calling *isNotThePowerOfTwo(),* which refer to [2].
- Cut the tail by removing the last element of input signal.
- When tested DFT and FFT by using a small file (Brass). The DFT took 10m, 46s, while the FFT only took 0.5s.
- In the DFT, we calculate every point of input signal. It takes long time.
- In the FFT algorithm, X(k) = G(k) + H(k) * W(k), we divided it into even part G(k) and odd part H(k). Because G(k) is periodic, G(k) = G(k + n), also H(k) is periodic, H(k) = H(k + n). So we only need to calculate X(k), and use G(k) and H(k) to calculate X(k + n). So FFT is much faster than IDFT.

Stage 3:

- Successfully implement the IFFT by these equation, $$IFFT(X) = \frac{1}{N} conj(FFT(conj(X)))$$

- Take the spectrum's conjugate. Take the FFT of that result. Then take the complex conjugate again. Get the real part of the final complex number, add into waveform.
- When tested IDFT and IFFT by using a small file (Brass). The IDFT took 4m, 47s, while the IFFT only took 0.4s.
- The IDFT calculates every point of input spectrum, it takes long time. While the IFFT use the forward FFT to get result, which is much faster than IDFT.

Stage 4:

Didn't complete the stage 4.

Reference:

- [1] https://algs4.cs.princeton.edu/99scientific/FFT.java.html
- [2] https://www.geeksforgeeks.org/java-program-to-find-whether-a-no-is-power-of-two/