**1. 목적**

- Discrete Fourier Transform에서 x[n]의 n값에 따른 그래프를 구현한다.

- FFT의 x[n]의 n값에 따른 그래프를 구현한다.

- 두 방식의 계산 시간을 ctime 함수를 이용하여 DFT와 FFT의 계산

**2. 변수**

- 라이브러리 : iostream, fstream, cmath 및 지정된 complex.h와 fft.h를 사용하였다

- 정의 : PI값을 3.141592로 지정하였다

- 함수 : main함수 내에서 모든 문제들을 입출력 txt 파일을 통해 지정하였다.

**3. 알고리즘(순서)**

- 라이브러리로 <iostream>, <fstream>, <math.h>를 받았다

- 제공받은 헤더파일 "complex.h"와 "FftD.h"를 라이브러리에 지정하였다

- PI를 3.141592로 지정했다

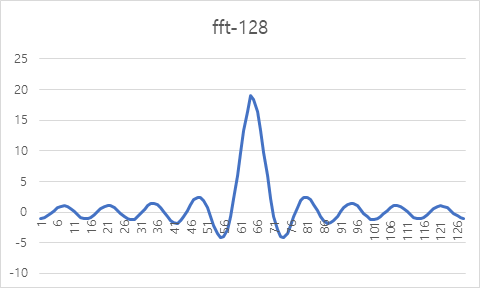
- main함수를 open하여 파일 출력을 위해 txt파일을 열었다.

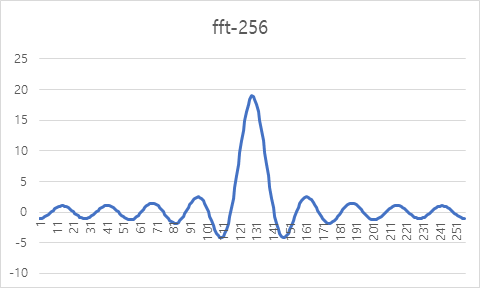
- ofstream을 이용하여 modulation된 신호의 DFT결과를 텍스트파일로 받아 엑셀로 그래프를 구현 해보았다.

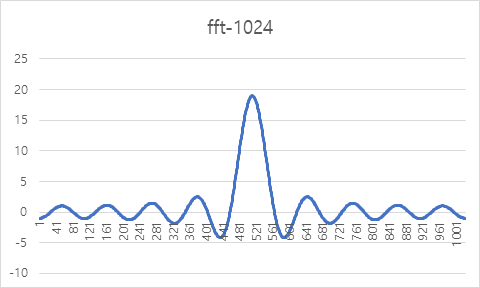
- ctime 함수를 이용하여 FFT와 DFT의 계산시간의 차이를 그래프로 그려보았다.

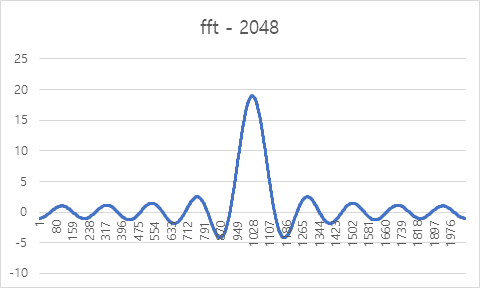
**4. FT 실행 결과(Assignment 1)**

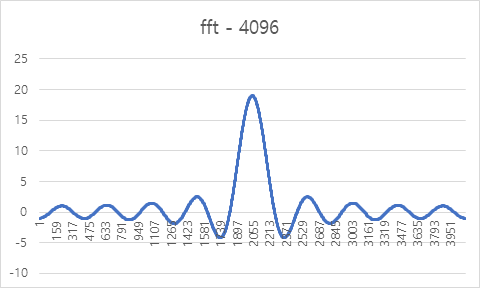
(1) square 함수

****

****

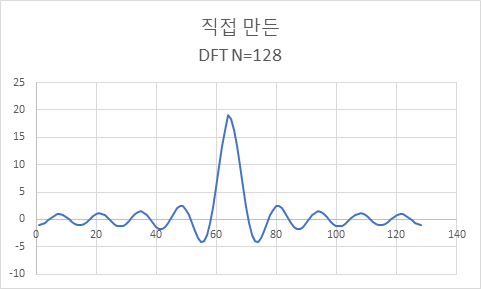
****

****

****

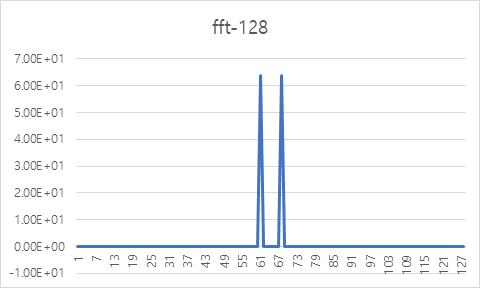
N값을 증가시키면서 Square함수의 FFT 결과를 분석한 결과 그래프의 큰 변화는 없었다.

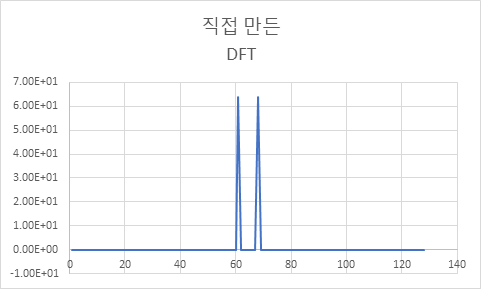
DFT 코드를 작성하여 N=128일 때 실행시켜본 결과,



FFT와 동일한 그래프를 도출하였다.

(2) cos 함수

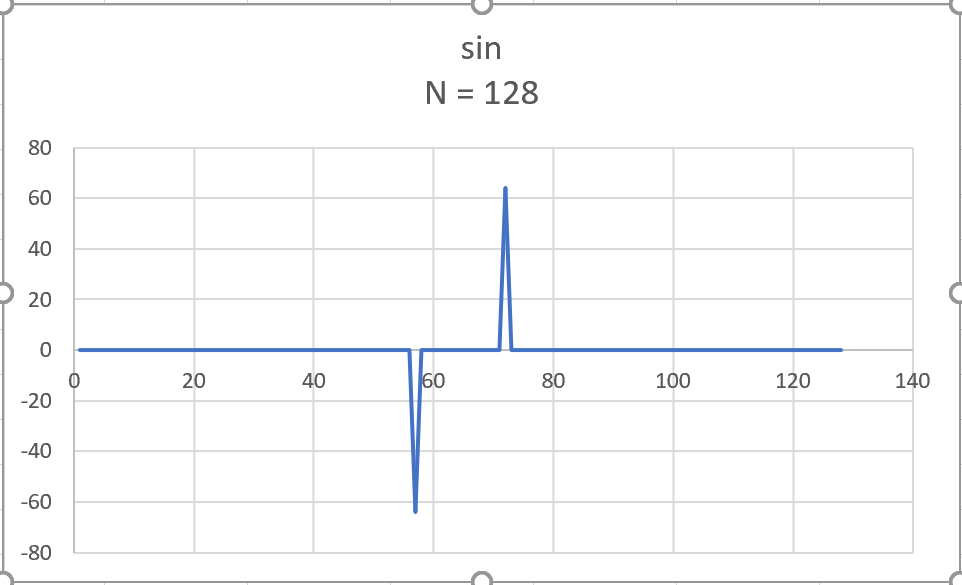
****

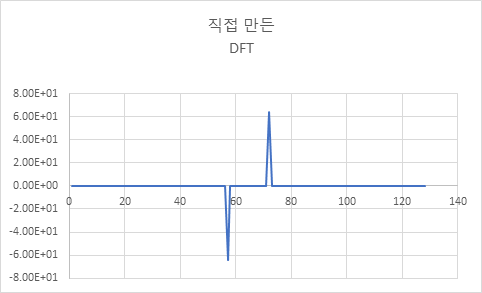
****

실행 결과, FFT와 DFT가 동일한 그래프임을 확인하였다.

두 개의 피크값을 갖는다.

(3) sin 함수

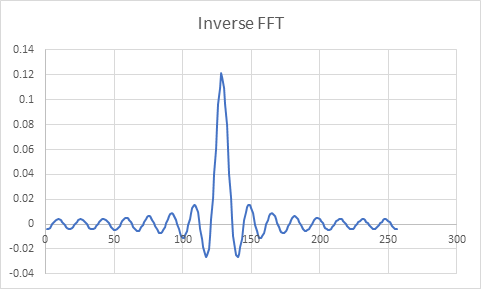
****

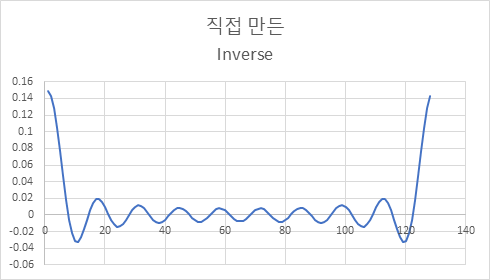


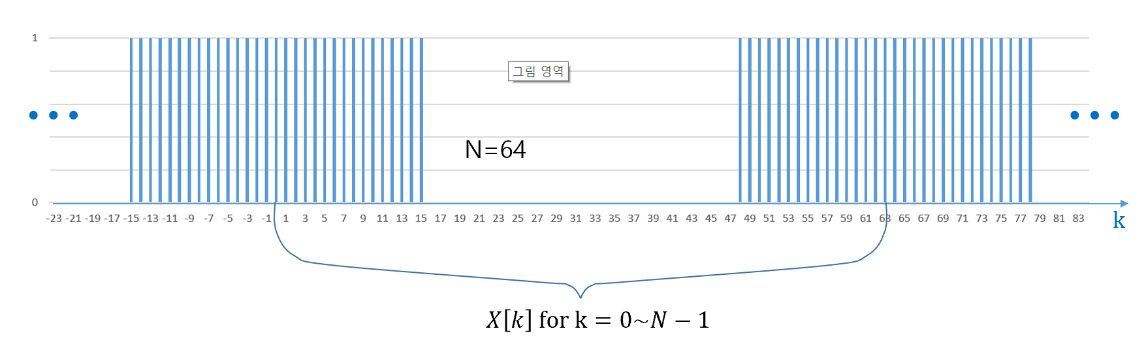
실행 결과, FFT와 DFT가 동일한 그래프임을 확인하였다.

두 개의 피크값을 갖는다.

**5. Inverse FT 실행 결과(Assignment 2)**

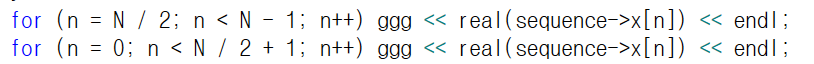
****

**(1) **

**IFFT와 다른 형태의 그래프를 DFT에서 얻게 되었는데, 이것은**

**0부터 다음 square의 중간까지 X[k]를 설정하는 방식으로 코드를 구현했기 때문이다.**

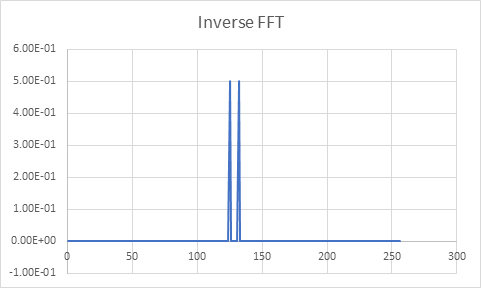
**IFFT의 코드에서**

****

**위와 같이 순서를 바꾸어 주어 이 문제를 해결하였는데**

**IDFT에서도 위의 코드를 적용시면 IFFT와 같은 그래프를 얻을 수 있다.**

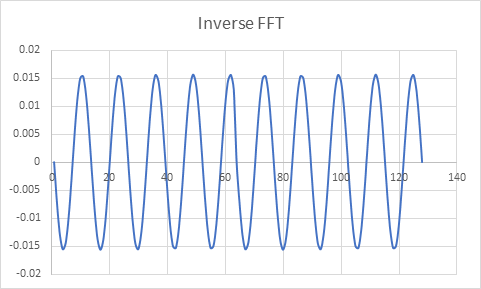
**(2)**

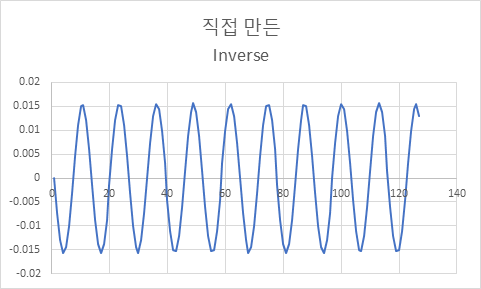
****

****

**square와 마찬가지로 순서를 바꾸어주면 IFFT와 같은 그래프를 얻을 수 있을 것이다.**

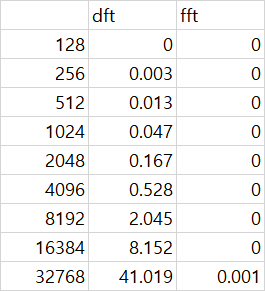
**(3)**

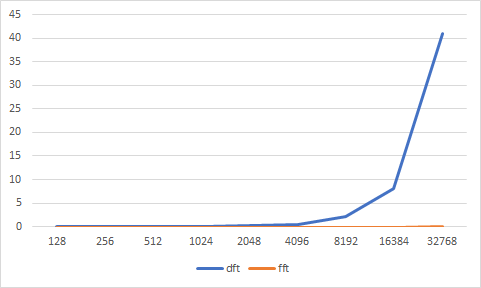
****

****

IFFT와 IDFT가 같은 것을 확인하였다.

**6. Complexity 비교**

FT의 square 함수에서 N값을 바꿔가며 dft와 fft를 실행하는데 걸리는 시간을 비교해보았다.



N값이 증가하면서 FFT의 계산시간은 거의 변화가 없었고 매우 짧았지만, DFT의 계산시간은 지수함수 꼴로 증가하는 것을 알 수 있었다.

N값이 커질수록 DFT와 FFT를 실행하는데 걸리는 시간 차이가 점점 벌어져서, 일정 N부터는 매우 극심한 차이를 갖는 것을 확인하였다.