Fast Fourier Transform

https://youtu.be/UdJMvSFqcG0

HANSUNG UNIVERSITY

CryptoCraft LAB

1. FFT란?

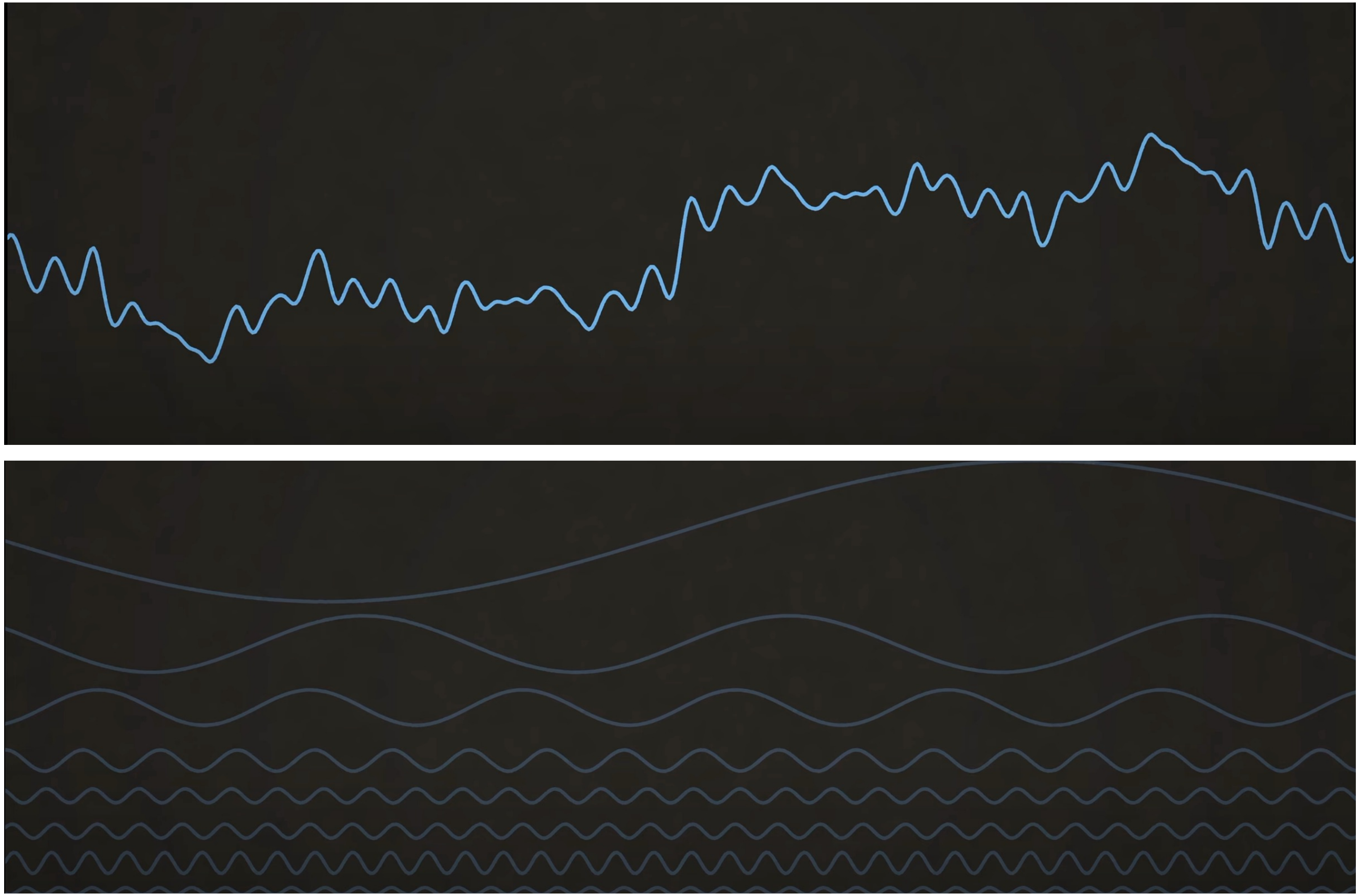
- Fast Fourier Transform
  - Discrete Fourier Transform을 빠르게 계산하기 위한 알고리즘
  - 많은 분야에서 활용되고 있는 알고리즘

- 개발 역사
  - 1950's 핵무기 개발을 위한 핵실험이 전 세계적으로 증가
    - 핵 실험으로 인한 방사능 피해
  - 1958년 핵무기 실험 중단을 위한 세계 회의 진행
    - 위 문제를 심각하게 받아들이고 있다는 증거로 핵실험 중단
      - 그러나 상대 국가가 핵실험을 안하고 있는지 알 수 없음(지하에서 실험할 경우)

- 핵실험과 지진을 구분하기 위해 Fortier transform 개발

1. FFT란?

- 문제는 Fourier transform의 방대한 양의 계산량이 문제



1. FFT란?

- DFT의 기본 공식

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N} \qquad k = 0, \dots, N-1,$$

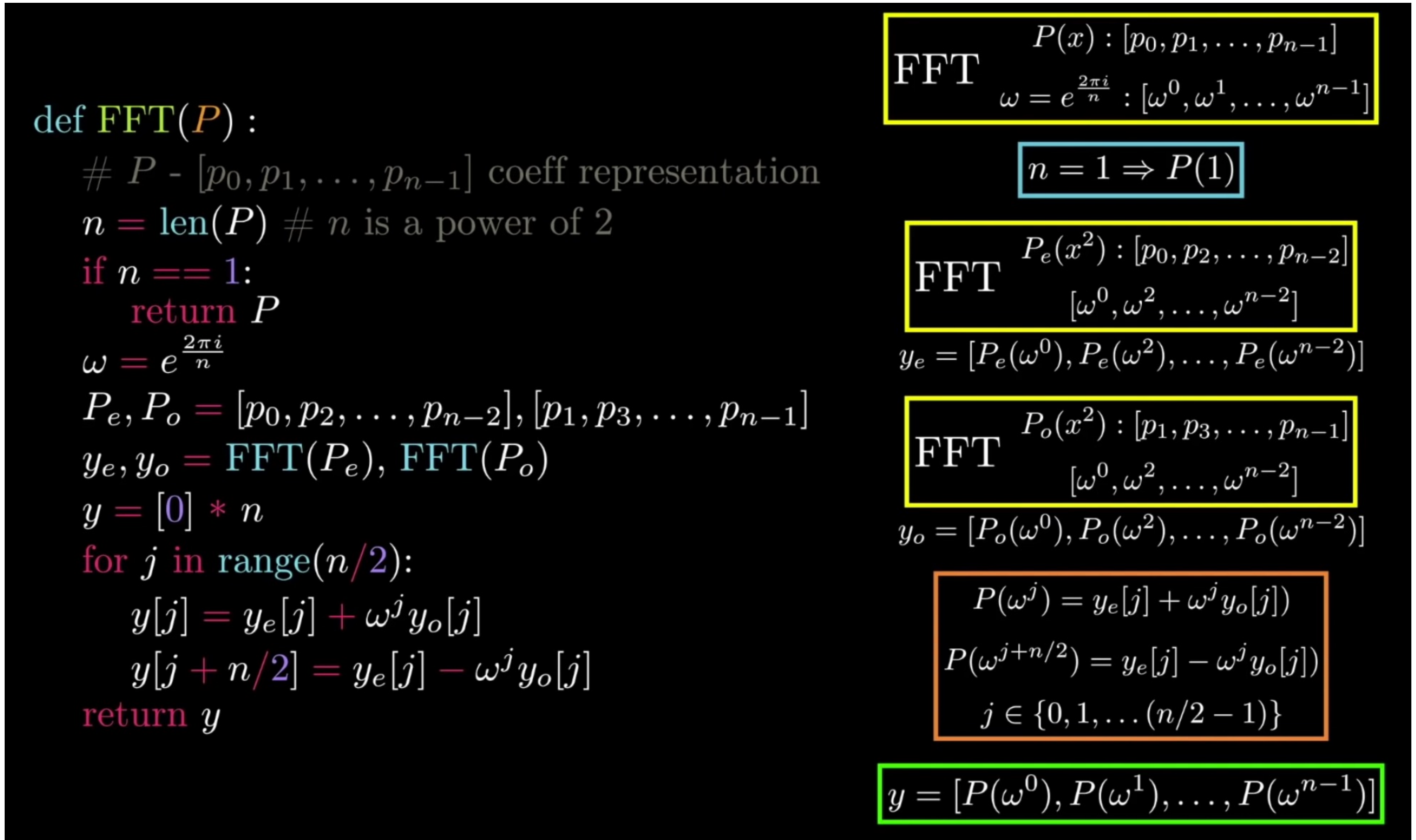
$$P(x) = p_0 + p_1 x + p_2 x^2 + \cdots + p_{n-1} x^{n-1}$$

$$\begin{bmatrix} P(\omega^0) \\ P(\omega^1) \\ P(\omega^2) \\ \vdots \\ P(\omega^{n-1}) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{bmatrix}}_{\text{Discrete Fourier Transform (DFT) matrix}} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_{n-1} \end{bmatrix}$$

$$x_k = \omega^k \text{ where } \omega = e^{\frac{2\pi i}{n}}$$

1. FFT란?

- FFT 알고리즘으로 인해 DFT의 계산 복잡도 $O(n^2)$을 $O(n\log n)$으로 감소
  - 30년 걸리는 계산이 30초만에 계산



2. NTT

$$P(x) = 3x^3 + 4x^2 + 4x + 1 \ / \ F_5$$

$$b_0 = P(2^0),\ b_1 = P(2^1),\ b_2 = P(2^2),\ b_3 = P(2^3)$$

$$P(2^0) = 3(2^0)^3 + 4(2^0)^2 + 4(2^0) + 1 = 12 \bmod 5 \equiv 2$$

$$P(2^1) = 3(2^1)^3 + 4(2^1)^2 + 4(2^1) + 1 = 49 \bmod 5 \equiv 4$$

$$P(2^2) = 3(2^2)^3 + 4(2^2)^2 + 4(2^2) + 1 = 273 \bmod 5 \equiv 3$$

2. NTT

$$P(x) = 3x^3 + 4x^2 + 4x + 1$$

$$b_0 = P(2^0),\ b_1 = P(2^1),\ b_2 = P(2^2),\ b_3 = P(2^3)$$

감 사 합 니 다