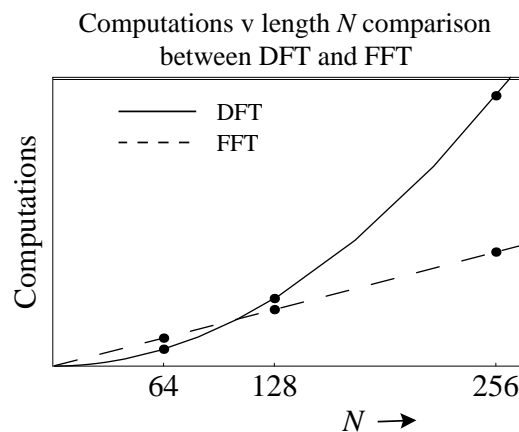


- Computing Time of FFT vs. the Direct DFT
- Algorithm for the FFT
- Implementation of the FFT

- Computing Time of FFT vs. the Direct DFT
- Computing order of the Direct DFT is  $N^2$ .
- Computing order of the Fast Fourier Transform is  $N \log N$ .



- Audio applications usually use analysis frames on the order of  $N=1000$ .
- FIR filter lengths are often hundreds of taps long.

## • Algorithm

- The length of the input must be a power of 2.
- Basic Idea is to split up the signal into pairs of numbers which have the following DFT.

$$\{a, b\} \leftrightarrow \{a + b, a - b\}$$

$$\text{dft}(\{a, b\}) = \sum_{n=0}^{N-1} x(n) e^{-j\pi nk} = \{ae^0 + be^0, ae^0 + be^{-j\pi}\} = \{a+b, a-b\}$$

- How a length N sequence is split up into pairs:

## • Implementation of the FFT

- Recursive method in LISP:

```
(defun RECURSIVE-FFT (x)
  (let ((N (length x)))
    (cond ((= N 2) (join (+ (elt x 0) (elt x 1))
                          (- (elt x 0) (elt x 1))))
          (t (let* ((x-even (RECURSIVE-FFT (even-elements-of x)))
                    (x-odd (mul (RECURSIVE-FFT (odd-elements-of x))
                                (roots-of-unity (/ N 2))))
                  (join (add x-even x-odd)
                        (sub x-even x-odd)))))))
```

- Bit-reversal method in C: