Fast Fourier Transform

Polynomials $p(x) = \sum_{i=0}^{n} a_i x^i$

```
EVALUATE(P[0..n], x):
  X ← 1        ⟨⟨X = x^j⟩⟩
  y ← 0
  for j ← 0 to n
      y ← y + P[j]·X
      X ← X·x
  return y
```

$O(n)$

```
ADD(P[0..n], Q[0..n]):
  for j ← 0 to n
      R[j] ← P[j] + Q[j]
  return R[0..n]
```

$O(n)$

```
MULTIPLY(P[0..n], Q[0..m]):
  for j ← 0 to n + m
      R[j] ← 0
  for j ← 0 to n
      for k ← 0 to m
          R[j+k] ← R[j+k] + P[j]·Q[k]
  return R[0..n+m]
```

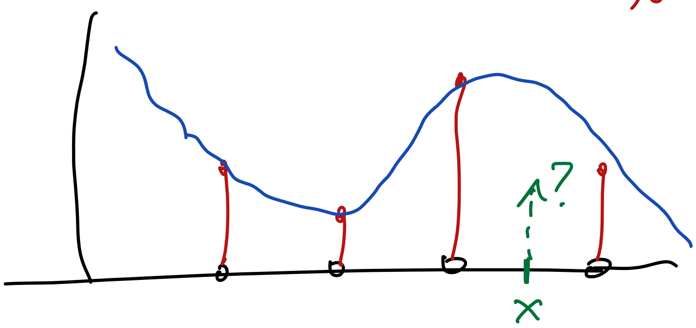
$O(n^2)$

Alternate representations:

- Roots: $p(x) = \sigma \cdot \prod_{i=1}^{n} (x - r_i)$

Eval: $O(n)$
Mult: $O(n)$
Add?

- Sample points Fix $x_0, x_1, \ldots x_n$ sample positions
representation is
$y_0\ y_1, \ldots y_n$ where $y_i = p(x_i)$



Add: $O(n)$ time
Mult: $O(n)$ time
Eval: $O(n^2)$ time

$$p(x) = \sum_{j=0}^{n-1} \left(\frac{y_j}{\prod_{k \neq j} (x_j - x_k)} \prod_{k \neq j} (x - x_k) \right)$$

Lagrange 17xx

Can we convert quickly?

representation	evaluate	add	multiply
coefficients	$O(n)$	$O(n)$	$\mathbf{O(n^2)}$
~~roots + scale~~	~~$O(n)$~~	~~∞~~	~~$O(n)$~~
samples	$\mathbf{O(n^2)}$	$O(n)$	$O(n)$

$O(n \log n)$ FFT

Converting coeffs $\rightarrow$ samples is a linear transformation

Vandermonde matrix

$O(n^2)$ time

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

Karatsuba:

x: | a | b |

y: | c | d |

$$p(x) = x \cdot P_{odd}(x^2) + P_{even}(x^2)$$

$$p(x) = x^3 + 3x^2 - 2x + 5 \quad \rightarrow \quad P_{odd} = x - 2, \quad P_{even} = 3x + 5$$

| 5 | -2 | 3 | 1 |

$\swarrow$ | 5 | 3 | $\qquad$ $\searrow$ | -2 | 1 |

X is collapsible if

- $|X| = 1$

-or-

- $X^2 = \{x^2 \mid x \in X\}$ is collapsible and has $|X|/2$ elements

Example: $\{1\}$

$X = \{-1, 1\} \Rightarrow X^2 = \{1\}$

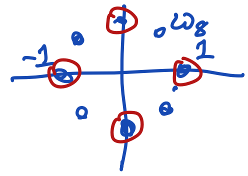
$T(n)$ = time to eval polynomial of degree n at collapsible set of n sample positions

$$T(n) = 2T(n/2) + O(n) = O(n \log n)$$

$$\{1\} \rightarrow \{-1, 1\} \rightarrow \{i, -i, -1, 1\} \rightarrow \left\{\pm\tfrac{\sqrt{2}}{2} \pm \tfrac{\sqrt{2}}{2} i, i, -i, -1, 1\right\} \rightarrow$$

$\boxed{2^k\text{th roots of unity}}$

$\omega_n = \cos\left(\frac{2\pi}{n}\right) + i \cdot \sin\left(\frac{2\pi}{n}\right)$



DFT of p:

$$\boxed{P^*[j] := p(\omega_n^j) = \sum_{k=0}^{n-1} P[k] \cdot \omega_n^{jk}}$$

<u>RADIX2FFT($P[0\,..\,n-1]$):</u>
    if $n = 1$
        return P

    for $j \leftarrow 0$ to $n/2 - 1$
        $U[j] \leftarrow P[2j]$ ← even
        $V[j] \leftarrow P[2j+1]$ ← odd

O(n)

    $U^* \leftarrow$ RADIX2FFT($U[0\,..\,n/2-1]$) ← $T(n/2)$
    $V^* \leftarrow$ RADIX2FFT($V[0\,..\,n/2-1]$) ← $T(n/2)$

    $\omega_n \leftarrow \cos(\frac{2\pi}{n}) + i \sin(\frac{2\pi}{n})$
    $\omega \leftarrow 1$

    for $j \leftarrow 0$ to $n/2 - 1$
        $P^*[j] \leftarrow U^*[j] + \omega \cdot V^*[j]$
        $P^*[j + n/2] \leftarrow U^*[j] - \omega \cdot V^*[j]$
        $\omega \leftarrow \omega \cdot \omega_n$

    return $P^*[0\,..\,n-1]$

O(n log n) time

```
FactorFFT(P[0 .. pq − 1]):
  ⟨⟨Copy/typecast to 2d array in row-major order⟩⟩
  for j ← 0 to p − 1
      for k ← 0 to q − 1
          A[j, k] ← P[jp + k]

  ⟨⟨Recursively apply order-p FFTs to columns⟩⟩
  for k ← 0 to q − 1
      B[·, k] ← FFT(A[·, k])

  ⟨⟨Multiply by twiddle factors⟩⟩
  for j ← 0 to p − 1
      for k ← 0 to q − 1
          B[·, k] ← B[·, k] · ω_pq^jk

  ⟨⟨Recursively apply order-q FFTs to rows⟩⟩
  for j ← 0 to p − 1
      C[j, ·] ← FFT(C[j, ·])

  ⟨⟨Copy/typecast to 1d array in column-major order⟩⟩
  for j ← 0 to p − 1
      for k ← 0 to q − 1
          P*[j + kq] ← C[j, k]

  return P*[0 .. pq − 1]
```

Figure A.2. The Gauss-Cooley-Tukey FFT algorithm

$$V = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \omega_n^3 & \cdots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \omega_n^6 & \cdots & \omega_n^{2(n-1)} \\ 1 & \omega_n^3 & \omega_n^6 & \omega_n^9 & \cdots & \omega_n^{3(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \omega_n^{3(n-1)} & \cdots & \omega_n^{(n-1)^2} \end{bmatrix}$$

$\boxed{\vec{y} = V\vec{a}}$

$\vec{a} = V^{-1}\vec{y}$

$\boxed{\textbf{Lemma A.1. } V^{-1} = \overline{V}/n}$

```
INVERSEFFT(P*[0..n−1]):
    P[0..n−1] ← FFT(P*)
    for j ← 0 to n−1
        P*[j] ← conj(P*[j])/n
    return P[0..n−1]
```

```
INVERSERADIX2FFT(P*[0..n−1]):
    if n = 1
        return P

    for j ← 0 to n/2−1
        U*[j] ← P*[2j]
        V*[j] ← P*[2j+1]

    U ← INVERSERADIX2FFT(U*[0..n/2−1])
    V ← INVERSERADIX2FFT(V*[0..n/2−1])

    conj(ω_n) ← cos(2π/n) − i sin(2π/n)
    conj(ω) ← 1

    for j ← 0 to n/2−1
        P[j]       ← (U[j] + conj(ω)·V[j])/2
        P[j + n/2] ← (U[j] − conj(ω)·V[j])/2
        conj(ω) ← conj(ω)·conj(ω_n)

    return P[0..n−1]
```

```
FFTMultiply(P[0..m-1], Q[0..n-1]):
    for j ← m to m+n-1
        P[j] ← 0
    for j ← n to m+n-1
        Q[j] ← 0
    P* ← FFT(P)
    Q* ← FFT(Q)
    for j ← 0 to m+n-1
        R*[j] ← P*[j] · Q*[j]
    return InverseFFT(R*)
```

$O(n \log n)$

Convolution

$a = \langle a_0, a_1, \ldots, a_{m-1} \rangle$ $\qquad$ $b = \langle b_0, b_1 \ldots b_{n-1} \rangle$

$$a * b = c = \langle c_0 \; c_1 \ldots c_{m+n-2} \rangle$$

where $c_k = \sum_{i+j=k} a_i b_j$