Discrete and Fast Fourier Transforms

Goal: Let $M(d)$ denote the number of binary bit operations needed to multiply two positive integers each with $\leq d$ bits.

Discrete and Fast Fourier Transforms

Goal: Let $M(d)$ denote the number of binary bit operations needed to multiply two positive integers each with $\leq d$ bits. We stated that $M(d) \ll d(\log d) \log\log d$. We give the basic idea behind the proof, not worrying so much about the running time but concentrating on the main idea of using fast Fourier transforms for performing multiplication.

Example. Let R be a rectangle, and suppose R is expressed as a union of rectangles R_j, $1 \leq j \leq r$, with edges parallel to R and common points only along these edges.

Discrete and Fast Fourier Transforms

Goal: Let $M(d)$ denote the number of binary bit operations needed to multiply two positive integers each with $\le d$ bits. We stated that $M(d) \ll d(\log d) \log\log d$. We give the basic idea behind the proof, not worrying so much about the running time but concentrating on the main idea of using fast Fourier transforms for performing multiplication.

Example. Let R be a rectangle, and suppose R is expressed as a union of rectangles R_j, $1 \le j \le r$, with edges parallel to R and common points only along these edges. Suppose further that each R_j has at least one edge of integer length.

Discrete and Fast Fourier Transforms

Goal: Let $M(d)$ denote the number of binary bit operations needed to multiply two positive integers each with $\leq d$ bits. We stated that $M(d) \ll d(\log d)\log\log d$. We give the basic idea behind the proof, not worrying so much about the running time but concentrating on the main idea of using fast Fourier transforms for performing multiplication.

Example. Let R be a rectangle, and suppose R is expressed as a union of rectangles R_j, $1 \leq j \leq r$, with edges parallel to R and common points only along these edges. Suppose further that each R_j has at least one edge of integer length. Then R itself has an edge of integer length.

Example. Let R be a rectangle, and suppose R is expressed as a union of rectangles R_j, $1 \leq j \leq r$, with edges parallel to R and common points only along these edges. Suppose further that each R_j has at least one edge of integer length. Then R itself has an edge of integer length.

$$\int_0^1 e^{i2\pi k\theta} d\theta = \frac{1}{2\pi}\int_0^{2\pi} e^{ik\theta} d\theta = \begin{cases} 0 & \text{if } k \in \mathbb{Z} - \{0\} \\ 1 & \text{if } k = 0 \end{cases}$$

$$\iint\limits_{R_j} e^{2\pi i(x+y)} dx\, dy = 0 \iff R_j \text{ has a side of integer length}$$

$$\left| \iint\limits_{R} e^{2\pi i(x+y)} dx\, dy \right| = \left| \sum_{j=1}^{r} \iint\limits_{R_j} e^{2\pi i(x+y)} dx\, dy \right|$$

Lemma. *Let n and k be integers with $n \geq 1$. Let $\omega = e^{2\pi i/n}$. Then*

$$\sum_{j=0}^{n-1} \omega^{kj} = \begin{cases} 0 & \textit{if } k \not\equiv 0 \pmod{n} \\ n & \textit{if } k \equiv 0 \pmod{n}. \end{cases}$$

Example. $1 - \frac{1}{4} + \frac{1}{7} - \frac{1}{10} + \cdots =$?

```
> evalf(sum((-1)^k/(3*k+1),k=0..10000));
      0.8356655135425729059601486843366923241711921020962
```

Lemma. *Let n and k be integers with $n \geq 1$. Let $\omega = e^{2\pi i/n}$. Then*

$$\sum_{j=0}^{n-1} \omega^{kj} = \begin{cases} 0 & \text{if } k \not\equiv 0 \pmod{n} \\ n & \text{if } k \equiv 0 \pmod{n}. \end{cases}$$

Example. $1 - \frac{1}{4} + \frac{1}{7} - \frac{1}{10} + \cdots =$?

```
> evalf(sum((-1)^k/(3*k+1),k=0..10000));
      0.8356655135425729059601486843366923241711921020962
> evalf(sum((-1)^k/(3*k+1),k=0..100000));
      0.8356505149174989051890324679305205761235311239980 6
```

Lemma. *Let n and k be integers with $n \geq 1$. Let $\omega = e^{2\pi i/n}$. Then*

$$\sum_{j=0}^{n-1} \omega^{kj} = \begin{cases} 0 & \text{if } k \not\equiv 0 \pmod{n} \\ n & \text{if } k \equiv 0 \pmod{n}. \end{cases}$$

Example. $1 - \frac{1}{4} + \frac{1}{7} - \frac{1}{10} + \cdots = ?$

```
> evalf(sum((-1)^k/(3*k+1),k=0..10000));
    0.83566551354257290596014868433669232417119210209620
> evalf(sum((-1)^k/(3*k+1),k=0..100000));
    0.83565051491749890518903246793052057612353112399806
> evalf(sum((-1)^k/(3*k+1),k=0..1000000));
    0.83564901493124883118895531926797084904818207352932
```

Lemma. *Let n and k be integers with $n \geq 1$. Let $\omega = e^{2\pi i/n}$. Then*

$$\sum_{j=0}^{n-1} \omega^{kj} = \begin{cases} 0 & \text{if } k \not\equiv 0 \pmod{n} \\ n & \text{if } k \equiv 0 \pmod{n}. \end{cases}$$

Example. $1 - \frac{1}{4} + \frac{1}{7} - \frac{1}{10} + \cdots = \ ?$

```
> evalf(sum((-1)^k/(3*k+1),k=0..10000));
    0.83566551354257290596014868433669232417119210209620
> evalf(sum((-1)^k/(3*k+1),k=0..100000));
    0.83565051491749890518903246793052057612353112399806
> evalf(sum((-1)^k/(3*k+1),k=0..1000000));
    0.83564901493124883118895531926797084904818207352932
> evalf((3*log(2)+sqrt(3)*Pi)/9);
    0.83564884826472105333710345970011076678652212748433
```

Lemma. *Let n and k be integers with $n \geq 1$. Let $\omega = e^{2\pi i/n}$. Then*

$$\sum_{j=0}^{n-1} \omega^{kj} = \begin{cases} 0 & \text{if } k \not\equiv 0 \pmod{n} \\ n & \text{if } k \equiv 0 \pmod{n}. \end{cases}$$

Example. $$1 - \frac{1}{4} + \frac{1}{7} - \frac{1}{10} + \cdots = \frac{3\log 2 + \sqrt{3}\pi}{9}$$

```
> evalf(sum((-1)^k/(3*k+1),k=0..10000));
    0.8356655135425729059601486843366923241711921020962 0
> evalf(sum((-1)^k/(3*k+1),k=0..100000));
    0.83565051491749890518903246793052057612353112399806
> evalf(sum((-1)^k/(3*k+1),k=0..1000000));
    0.83564901493124883118895531926797084904818207352932
> evalf((3*log(2)+sqrt(3)*Pi)/9);
    0.83564884826472105333710345970011076678652212748433
```

$$\mathbf{1 - \frac{1}{4} + \frac{1}{7} - \frac{1}{10} + \cdots = \frac{3\log 2 + \sqrt{3}\pi}{9}}$$

Call the sum $\boldsymbol{S}$.

$$1 - \frac{1}{4} + \frac{1}{7} - \frac{1}{10} + \cdots = \frac{3\log 2 + \sqrt{3}\pi}{9}$$

Call the sum S. We use that

$$z^2 \log(1+z) = z^3 - \frac{z^4}{2} + \frac{z^5}{3} - \frac{z^6}{4} + \cdots .$$

$$1-\frac{1}{4}+\frac{1}{7}-\frac{1}{10}+\cdots=\frac{3\log 2+\sqrt{3}\pi}{9}$$

Call the sum S. We use that

$$z^2\log(1+z)=z^3-\frac{z^4}{2}+\frac{z^5}{3}-\frac{z^6}{4}+\cdots.$$

Let $\omega=e^{2\pi i/3}$, and note that

$$(*)\qquad (1+\omega)(1+\omega^2)=1+\omega+\omega^2+\omega^3=\omega^3=1.$$

$$1-\frac{1}{4}+\frac{1}{7}-\frac{1}{10}+\cdots=\frac{3\log 2+\sqrt{3}\pi}{9}$$

Call the sum S. We use that

$$z^2\log(1+z)=z^3-\frac{z^4}{2}+\frac{z^5}{3}-\frac{z^6}{4}+\cdots.$$

Let $\omega=e^{2\pi i/3}$, and note that

$$(*)\qquad (1+\omega)(1+\omega^2)=1+\omega+\omega^2+\omega^3=\omega^3=1.$$

Also,

$$(**)\qquad \omega=\frac{-1+\sqrt{3}\,i}{2}\quad\text{and}\quad \omega^2=\frac{-1-\sqrt{3}\,i}{2}.$$

Lemma. *Let n and k be integers with $n \geq 1$. Let $\omega = e^{2\pi i/n}$. Then*

$$\sum_{j=0}^{n-1} \omega^{kj} = \begin{cases} 0 & \text{if } k \not\equiv 0 \pmod{n} \\ n & \text{if } k \equiv 0 \pmod{n}. \end{cases}$$

$$z^2 \log(1+z) = z^3 - \frac{z^4}{2} + \frac{z^5}{3} - \frac{z^6}{4} + \cdots .$$

Let $\omega = e^{2\pi i/3}$, and note that

$$(*) \qquad (1+\omega)(1+\omega^2) = 1 + \omega + \omega^2 + \omega^3 = \omega^3 = 1.$$

Also,

$$(**) \qquad \omega = \frac{-1+\sqrt{3}\, i}{2} \quad \text{and} \quad \omega^2 = \frac{-1-\sqrt{3}\, i}{2}.$$

The lemma implies that

$$3S = \log(1+1) + \omega^2 \log(1+\omega) + \omega \log(1+\omega^2).$$

$$1-\frac{1}{4}+\frac{1}{7}-\frac{1}{10}+\cdots=\frac{3\log 2+\sqrt{3}\pi}{9}$$

Let $\omega = e^{2\pi i/3}$, and note that

$$(*) \qquad (1+\omega)(1+\omega^2) = 1+\omega+\omega^2+\omega^3 = \omega^3 = 1.$$

Also,

$$(**) \qquad \omega = \frac{-1+\sqrt{3}\,i}{2} \quad \text{and} \quad \omega^2 = \frac{-1-\sqrt{3}\,i}{2}.$$

The lemma implies that

$$3S = \log(1+1) + \omega^2\log(1+\omega) + \omega\log(1+\omega^2).$$

From $(*)$ and $(**)$,

$$3S = \log 2 + \sqrt{3}\,i\log\left(1+\omega^2\right)$$

$$1-\frac{1}{4}+\frac{1}{7}-\frac{1}{10}+\cdots=\frac{3\log 2+\sqrt{3}\pi}{9}$$

Also,

$$(**) \qquad \omega=\frac{-1+\sqrt{3}\,i}{2} \quad \text{and} \quad \omega^2=\frac{-1-\sqrt{3}\,i}{2}.$$

The lemma implies that

$$3S=\log(1+1)+\omega^2\log(1+\omega)+\omega\log(1+\omega^2).$$

From (*) and (**),

$$\begin{aligned}3S&=\log 2+\sqrt{3}\,i\log\left(1+\omega^2\right)\\&=\log 2+\sqrt{3}i\log\left(-\omega\right)\end{aligned}$$

$$1-\frac{1}{4}+\frac{1}{7}-\frac{1}{10}+\cdots=\frac{3\log 2+\sqrt{3}\pi}{9}$$

Also,

$$(**)\qquad \omega=\frac{-1+\sqrt{3}\,i}{2}\quad\text{and}\quad \omega^2=\frac{-1-\sqrt{3}\,i}{2}.$$

The lemma implies that

$$3S=\log(1+1)+\omega^2\log(1+\omega)+\omega\log(1+\omega^2).$$

From $(*)$ and $(**)$,

$$\begin{aligned}3S&=\log 2+\sqrt{3}\,i\log\left(1+\omega^2\right)\\&=\log 2+\sqrt{3}i\log\left(-\omega\right)\\&=\log 2+\sqrt{3}\,\pi/3,\end{aligned}$$

from which the value of S above follows.

Definitions and notations.

Definitions and notations. For n a positive integer, we set $\omega = \omega_n = e^{2\pi i/n}$.

Definitions and notations. For n a positive integer, we set $\omega = \omega_n = e^{2\pi i/n}$. Let

$$D = D(n, \omega) = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{pmatrix}.$$

Definitions and notations. For n a positive integer, we set $\omega = \omega_n = e^{2\pi i/n}$. Let

$$D = D(n,\omega) = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{pmatrix}.$$

For $\vec{u} = \langle u_0, u_1, \ldots, u_{n-1} \rangle \in \mathbb{C}^n$, define $\vec{v} = \langle v_0, v_1, \ldots, v_{n-1} \rangle$, called the discrete Fourier transform of $\vec{u}$, by $\vec{v} = D\,\vec{u}^T$.

Definitions and notations. For n a positive integer, we set $\omega = \omega_n = e^{2\pi i/n}$. Let

$$D = D(n,\omega) = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{pmatrix}.$$

For $\vec{u} = \langle u_0, u_1, \ldots, u_{n-1} \rangle \in \mathbb{C}^n$, define $\vec{v} = \langle v_0, v_1, \ldots, v_{n-1} \rangle$, called the discrete Fourier transform of $\vec{u}$, by $\vec{v} = D\,\vec{u}^T$. The inverse discrete Fourier transform of a vector $\vec{v} \in \mathbb{C}^n$ is defined as $(1/n)D(n,\omega^{-1})\,\vec{v}^T$.

Why is $(1/n)D(n,\omega^{-1})D(n,\omega)$ the identity matrix?

Lemma. *Let n and k be integers with $n \geq 1$. Let $\omega = e^{2\pi i/n}$. Then*

$$\sum_{j=0}^{n-1} \omega^{kj} = \begin{cases} 0 & \text{if } k \not\equiv 0 \pmod{n} \\ n & \text{if } k \equiv 0 \pmod{n}. \end{cases}$$

Definitions and notations. For n a positive integer, we set $\omega = \omega_n = e^{2\pi i/n}$. Let

$$D = D(n, \omega) = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{pmatrix}.$$

For $\vec{u} = \langle u_0, u_1, \ldots, u_{n-1} \rangle \in \mathbb{C}^n$, define $\vec{v} = \langle v_0, v_1, \ldots, v_{n-1} \rangle$, called the discrete Fourier transform of $\vec{u}$, by $\vec{v} = D\,\vec{u}^T$. The inverse discrete Fourier transform of a vector $\vec{v} \in \mathbb{C}^n$ is defined as $(1/n)D(n, \omega^{-1})\,\vec{v}^T$.

Why is $(1/n)D(n, \omega^{-1})D(n, \omega)$ the identity matrix?

A Polynomial Connection

Observe that if $f(x) = \sum_{j=1}^{n-1} a_j x^j \in \mathbb{C}[x]$, **then**

$$D \langle a_0, a_1, \ldots, a_{n-1} \rangle^T = \text{ ?}$$

$$D = D(n, \omega) = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{pmatrix}$$

A Polynomial Connection

Observe that if $f(x) = \sum_{j=1}^{n-1} a_j x^j \in \mathbb{C}[x]$, then

$$D \langle a_0, a_1, \ldots, a_{n-1} \rangle^T = \langle f(1), f(\omega), \ldots, f(\omega^{n-1}) \rangle^T.$$

$$D = D(n, \omega) = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{pmatrix}$$

A Polynomial Connection

Observe that if $f(x) = \sum_{j=1}^{n-1} a_j x^j \in \mathbb{C}[x]$, then

$$D \langle a_0, a_1, \ldots, a_{n-1} \rangle^T = \langle f(1), f(\omega), \ldots, f(\omega^{n-1}) \rangle^T.$$

On the other hand, if we know $f(1), f(\omega), \ldots, f(\omega^{n-1})$ and do not know the coefficients of $f(x)$, then we can obtain the coefficients from

$$D^{-1} \langle f(1), f(\omega), \ldots, f(\omega^{n-1}) \rangle^T = \langle a_0, a_1, \ldots, a_{n-1} \rangle^T.$$

$$D = D(n, \omega) = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{pmatrix}$$

A Polynomial Connection

Observe that if $f(x) = \sum_{j=1}^{n-1} a_j x^j \in \mathbb{C}[x]$, then

$$D\,\langle a_0, a_1, \ldots, a_{n-1}\rangle^T = \langle f(1), f(\omega), \ldots, f(\omega^{n-1})\rangle^T.$$

On the other hand, if we know $f(1), f(\omega), \ldots, f(\omega^{n-1})$ and do not know the coefficients of $f(x)$, then we can obtain the coefficients from

$$D^{-1}\langle f(1), f(\omega), \ldots, f(\omega^{n-1})\rangle^T = \langle a_0, a_1, \ldots, a_{n-1}\rangle^T.$$

Note that here $D^{-1} = (1/n)D(n, \omega^{-1})$.

$$D = D(n,\omega) = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{pmatrix}$$

A Polynomial Connection

Observe that if $f(x) = \sum_{j=1}^{n-1} a_j x^j \in \mathbb{C}[x]$, then

$$D\,\langle a_0, a_1, \ldots, a_{n-1}\rangle^T = \langle f(1), f(\omega), \ldots, f(\omega^{n-1})\rangle^T.$$

On the other hand, if we know $f(1), f(\omega), \ldots, f(\omega^{n-1})$ and do not know the coefficients of $f(x)$, then we can obtain the coefficients from

$$D^{-1}\langle f(1), f(\omega), \ldots, f(\omega^{n-1})\rangle^T = \langle a_0, a_1, \ldots, a_{n-1}\rangle^T.$$

Note that here $D^{-1} = (1/n)D(n, \omega^{-1})$.

If

$$F(x) = f(1) + f(\omega)x + \cdots + f(w^{n-1})x^{n-1},$$

then

$$F(1) = na_0,\ F\left(\omega^{-1}\right) = na_1,\ \ldots, F\left(\omega^{-(n-1)}\right) = na_{n-1}.$$

The Fast Fourier Transform

We explain a fast way of performing the computation in

$$(*) \qquad \boldsymbol{D} \left\langle \boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{n-1} \right\rangle^T = \left\langle \boldsymbol{f}(1), \boldsymbol{f}(\boldsymbol{\omega}), \ldots, \boldsymbol{f}(\boldsymbol{\omega}^{n-1}) \right\rangle^T.$$

The Fast Fourier Transform

We explain a fast way of performing the computation in

$$(*) \qquad \boldsymbol{D \langle a_0, a_1, \ldots, a_{n-1} \rangle^T = \langle f(1), f(\omega), \ldots, f(\omega^{n-1}) \rangle^T}.$$

There exist unique polynomials $\boldsymbol{f_e}$ and $\boldsymbol{f_o}$ in $\mathbb{C}[\boldsymbol{x}]$ such that $\boldsymbol{f(x) = f_e(x^2) + x f_o(x^2)}$.

The Fast Fourier Transform

We explain a fast way of performing the computation in

$$(*) \qquad D\,\langle a_0, a_1, \ldots, a_{n-1}\rangle^T = \langle f(1), f(\omega), \ldots, f(\omega^{n-1})\rangle^T.$$

There exist unique polynomials f_e and f_o in $\mathbb{C}[x]$ such that $f(x) = f_e(x^2) + xf_o(x^2)$. Calculate $f_e(\omega^{2j})$, $f_o(\omega^{2j})$ and $\omega^j f_o(\omega^{2j})$ for $0 \le j \le n-1$ to obtain the right side of $(*)$.

Why is this Fast?

For convenience, view n as a power of 2.

The Fast Fourier Transform

We explain a fast way of performing the computation in

$$(*) \qquad D\,\langle a_0, a_1, \ldots, a_{n-1}\rangle^T = \langle f(1), f(\omega), \ldots, f(\omega^{n-1})\rangle^T.$$

There exist unique polynomials f_e and f_o in $\mathbb{C}[x]$ such that $f(x) = f_e(x^2) + x f_o(x^2)$. Calculate $f_e(\omega^{2j})$, $f_o(\omega^{2j})$ and $\omega^j f_o(\omega^{2j})$ for $0 \le j \le n-1$ to obtain the right side of $(*)$.

Why is this Fast?

For convenience, view n as a power of 2. Let $A(n)$ be the number of arithmetic operations that one needs to compute $f(1), f(\omega), \ldots, f(\omega^{n-1})$.

The Fast Fourier Transform

We explain a fast way of performing the computation in

$$(*) \qquad D\,\langle a_0, a_1, \ldots, a_{n-1}\rangle^T = \langle f(1), f(\omega), \ldots, f(\omega^{n-1})\rangle^T.$$

There exist unique polynomials f_e and f_o in $\mathbb{C}[x]$ such that $f(x) = f_e(x^2) + x f_o(x^2)$. Calculate $f_e(\omega^{2j})$, $f_o(\omega^{2j})$ and $\omega^j f_o(\omega^{2j})$ for $0 \le j \le n-1$ to obtain the right side of $(*)$.

Why is this Fast?

For convenience, view n as a power of 2. Let $A(n)$ be the number of arithmetic operations that one needs to compute $f(1), f(\omega), \ldots, f(\omega^{n-1})$. Computing in an obvious way gives $A(n) \le n^2 + n(n-1)$.

The Fast Fourier Transform

We explain a fast way of performing the computation in

$$(*) \qquad D\,\langle a_0, a_1, \ldots, a_{n-1}\rangle^T = \langle f(1), f(\omega), \ldots, f(\omega^{n-1})\rangle^T.$$

There exist unique polynomials f_e and f_o in $\mathbb{C}[x]$ such that $f(x) = f_e(x^2) + x f_o(x^2)$. Calculate $f_e(\omega^{2j})$, $f_o(\omega^{2j})$ and $\omega^j f_o(\omega^{2j})$ for $0 \leq j \leq n-1$ to obtain the right side of $(*)$.

Why is this Fast?

For convenience, view n as a power of 2. Let $A(n)$ be the number of arithmetic operations that one needs to compute $f(1), f(\omega), \ldots, f(\omega^{n-1})$. Computing in an obvious way gives $A(n) \leq n^2 + n(n-1)$. Observe that

$$f_e(\omega_n^{2j}) = f_e(\omega_n^{2(j+(n/2))}) \qquad \text{for } 0 \leq j \leq (n/2) - 1.$$

$$(*) \qquad D\,\langle a_0, a_1, \ldots, a_{n-1}\rangle^T = \langle f(1), f(\omega), \ldots, f(\omega^{n-1})\rangle^T.$$

There exist unique polynomials f_e and f_o in $\mathbb{C}[x]$ such that $f(x) = f_e(x^2) + x f_o(x^2)$. Calculate $f_e(\omega^{2j})$, $f_o(\omega^{2j})$ and $\omega^j f_o(\omega^{2j})$ for $0 \le j \le n-1$ to obtain the right side of $(*)$.

For convenience, view n as a power of 2. Let $A(n)$ be the number of arithmetic operations that one needs to compute $f(1), f(\omega), \ldots, f(\omega^{n-1})$. Computing in an obvious way gives $A(n) \le n^2 + n(n-1)$. Observe that

$$f_e(\omega_n^{2j}) = f_e(\omega_n^{2(j+(n/2))}) \qquad \text{for } 0 \le j \le (n/2) - 1.$$

Similar equations hold with f_e replaced by f_o.

$$(*) \qquad \boldsymbol{D}\,\langle a_0, a_1, \ldots, a_{n-1}\rangle^T = \langle f(1), f(\omega), \ldots, f(\omega^{n-1})\rangle^T.$$

There exist unique polynomials f_e and f_o in $\mathbb{C}[x]$ such that $f(x) = f_e(x^2) + x f_o(x^2)$. Calculate $f_e(\omega^{2j})$, $f_o(\omega^{2j})$ and $\omega^j f_o(\omega^{2j})$ for $0 \le j \le n-1$ to obtain the right side of $(*)$.

For convenience, view n as a power of 2. Let $A(n)$ be the number of arithmetic operations that one needs to compute $f(1), f(\omega), \ldots, f(\omega^{n-1})$. Computing in an obvious way gives $A(n) \le n^2 + n(n-1)$. Observe that

$$f_e(\omega_n^{2j}) = f_e(\omega_n^{2(j+(n/2))}) \qquad \text{for } 0 \le j \le (n/2) - 1.$$

Similar equations hold with f_e replaced by f_o. We deduce that computing $f_e(\omega^{2j})$ and $f_o(\omega^{2j})$ takes $2A(n/2)$ arithmetic operations.

$$(*) \qquad D\,\langle a_0, a_1, \ldots, a_{n-1}\rangle^T = \langle f(1), f(\omega), \ldots, f(\omega^{n-1})\rangle^T.$$

There exist unique polynomials f_e and f_o in $\mathbb{C}[x]$ such that $f(x) = f_e(x^2) + xf_o(x^2)$. Calculate $f_e(\omega^{2j})$, $f_o(\omega^{2j})$ and $\omega^j f_o(\omega^{2j})$ for $0 \leq j \leq n-1$ to obtain the right side of $(*)$.

For convenience, view n as a power of 2. Let $A(n)$ be the number of arithmetic operations that one needs to compute $f(1), f(\omega), \ldots, f(\omega^{n-1})$. Computing in an obvious way gives $A(n) \leq n^2 + n(n-1)$. Observe that

$$f_e(\omega_n^{2j}) = f_e(\omega_n^{2(j+(n/2))}) \qquad \text{for } 0 \leq j \leq (n/2) - 1.$$

Similar equations hold with f_e replaced by f_o. We deduce that computing $f_e(\omega^{2j})$ and $f_o(\omega^{2j})$ takes $2A(n/2)$ arithmetic operations. Multiplying $f_o(\omega^{2j})$ by ω^j and then adding $f_e(\omega^{2j})$ takes $2n$ more arithmetic operations.

$$(*) \qquad D\,\langle a_0, a_1, \ldots, a_{n-1}\rangle^T = \langle f(1), f(\omega), \ldots, f(\omega^{n-1})\rangle^T.$$

There exist unique polynomials f_e and f_o in $\mathbb{C}[x]$ such that $f(x) = f_e(x^2) + x f_o(x^2)$. Calculate $f_e(\omega^{2j})$, $f_o(\omega^{2j})$ and $\omega^j f_o(\omega^{2j})$ for $0 \leq j \leq n-1$ to obtain the right side of $(*)$.

For convenience, view n as a power of 2. Let $A(n)$ be the number of arithmetic operations that one needs to compute $f(1), f(\omega), \ldots, f(\omega^{n-1})$. Computing in an obvious way gives $A(n) \leq n^2 + n(n-1)$. Observe that

$$f_e(\omega_n^{2j}) = f_e(\omega_n^{2(j+(n/2))}) \qquad \text{for } 0 \leq j \leq (n/2) - 1.$$

Similar equations hold with f_e replaced by f_o. We deduce that computing $f_e(\omega^{2j})$ and $f_o(\omega^{2j})$ takes $2A(n/2)$ arithmetic operations. Multiplying $f_o(\omega^{2j})$ by ω^j and then adding $f_e(\omega^{2j})$ takes $2n$ more arithmetic operations. We obtain

$$A(n) = 2A(n/2) + 2n \implies$$

$$(*) \qquad D\,\langle a_0, a_1, \ldots, a_{n-1}\rangle^T = \langle f(1), f(\omega), \ldots, f(\omega^{n-1})\rangle^T.$$

There exist unique polynomials f_e and f_o in $\mathbb{C}[x]$ such that $f(x) = f_e(x^2) + x f_o(x^2)$. Calculate $f_e(\omega^{2j})$, $f_o(\omega^{2j})$ and $\omega^j f_o(\omega^{2j})$ for $0 \le j \le n-1$ to obtain the right side of $(*)$.

For convenience, view n as a power of 2. Let $A(n)$ be the number of arithmetic operations that one needs to compute $f(1), f(\omega), \ldots, f(\omega^{n-1})$. Computing in an obvious way gives $A(n) \le n^2 + n(n-1)$. Observe that

$$f_e(\omega_n^{2j}) = f_e(\omega_n^{2(j+(n/2))}) \qquad \text{for } 0 \le j \le (n/2) - 1.$$

Similar equations hold with f_e replaced by f_o. We deduce that computing $f_e(\omega^{2j})$ and $f_o(\omega^{2j})$ takes $2A(n/2)$ arithmetic operations. Multiplying $f_o(\omega^{2j})$ by ω^j and then adding $f_e(\omega^{2j})$ takes $2n$ more arithmetic operations. We obtain

$$A(n) = 2A(n/2) + 2n \implies A(n) = (2/\log 2)\, n \log n + Cn$$

for some constant C.

Math 788: Computational Number Theory

- Math 788 Computational Number Theory Syllabus
- Math 788 Computational Number Theory Course Description
- Math 788 Computational Number Theory Class Notes
- Lectures on Primality Testing in Polynomial Time
- Computational Material on Polynomials
- Lectures

Lecture 1	Lecture 2	Lecture 3	Lecture 4	Lecture 5
Lecture 6	Lecture 7	Lecture 8	Lecture 9	Lecture 10
Lecture 11	Lecture 12	Lecture 13	Lecture 14	Lecture 15
Lecture 16	Lecture 17	Lecture 18	Lecture 19	Lecture 20
Lecture 21	Lecture 22	Lecture 23	Lecture 24	Lecture 25
Lecture 26				

- Test Materials

  Review List
  Test from 2007
  Final Exam from 2007
  Old Comp Exam Problems

- Graduate Number Theory Courses At The University of South Carolina

- Lectures on Primality Testing in Polynomial Time
- Computational Material on Polynomials
- **Lectures**

Lecture 1	Lecture 2	Lecture 3	Lecture 4	Lecture 5
Lecture 6	Lecture 7	Lecture 8	Lecture 9	Lecture 10
Lecture 11	Lecture 12	Lecture 13	Lecture 14	Lecture 15
Lecture 16	Lecture 17	Lecture 18	Lecture 19	Lecture 20
Lecture 21	Lecture 22	Lecture 23	Lecture 24	Lecture 25
Lecture 26				

- **Test Materials**

  Review List

  Test from 2007

  Final Exam from 2007

  Old Comp Exam Problems

- Graduate Number Theory Courses At The University of South Carolina

3. Be able to define a strong pseudoprime to the base b and be able to prove that no n is a strong pseudoprime to every base b with $1 \leq b \leq n$ and $\gcd(b, n) = 1$.

not directly related to computational aspects of number theory such as topics in Elementary

5. Be able to state and prove the Proth, Pocklington, Lehmer Test for primality.

6. Be able to prove that most numbers n have a prime factor $> \sqrt{n}$.

(This is not a sure prime test.)

8. Be able to factor n using Dixon's Algorithm (see homework).

7. Be able to explain Pollard's ρ-Algorithm including Floyd's cycle-finding algorithm

9. Be able to factor n using the Quadratic Sieve Algorithm.

9. Be able to factor n using the Quadratic Sieve Algorithm. (You will be given some informa-

10. Be able to prove Landau's inequality for the size of the factors of a polynomial.

11. Let $f(x) = \sum_{j=1}^{n-1} a_j x^j \in \mathbb{C}[x]$ and recall (7) from the notes which reads

12. Be able to state and prove Hadamard's inequality.

13. Be able to define what it means for a basis in a lattice to be *reduced*.

14. Be able to show (10) in the notes, that is that

14. Be able to prove $\mathbf{b} \in \mathcal{L},\ \mathbf{b} \neq \mathbf{0} \implies \|\mathbf{b}_1\| \leq 2^{(n-1)/2} \|\mathbf{b}\|$.

You have the power to request and even make changes to the list above. I have the power to veto the changes.

1. Be able to do problems related to any of the ~~homework~~. **Practice Problems**

3. Be able to define a strong pseudoprime to the base b and be able to prove that no n is a strong pseudoprime to every base b with $1 \leq b \leq n$ and $\gcd(b, n) = 1$.

2. Be able to prove that $\gcd(u, v)$ is $\asymp \log N$ on average and that usually it's much smaller.

5. Be able to state and prove the Proth, Pocklington, Lehmer Test for primality.

6. Be able to prove that most numbers n have a prime factor $> \sqrt{n}$.

8. Be able to factor n using Dixon's Algorithm (see homework).

9. Be able to factor n using the Quadratic Sieve Algorithm.

tion and you will need to make use of it and give the remaining details.)

10. Be able to prove Landau's inequality for the size of the factors of a polynomial.

12. Be able to state and prove Hadamard's inequality.

13. Be able to define what it means for a basis in a lattice to be *reduced*.

14. Be able to show (10) in the notes, that is that

14. Be able to prove $\mathbf{b} \in \mathcal{L},\ \mathbf{b} \neq \mathbf{0} \implies \|\mathbf{b}_1\| \leq 2^{(n-1)/2}\|\mathbf{b}\|$.

- Lectures on Primality Testing in Polynomial Time
- Computational Material on Polynomials
- **Lectures**

Lecture 1	Lecture 2	Lecture 3	Lecture 4	Lecture 5
Lecture 6	Lecture 7	Lecture 8	Lecture 9	Lecture 10
Lecture 11	Lecture 12	Lecture 13	Lecture 14	Lecture 15
Lecture 16	Lecture 17	Lecture 18	Lecture 19	Lecture 20
Lecture 21	Lecture 22	Lecture 23	Lecture 24	Lecture 25
Lecture 26				

- **Test Materials**

  Review List
  Test from 2007
  Final Exam from 2007
  Old Comp Exam Problems

- Graduate Number Theory Courses At The University of South Carolina