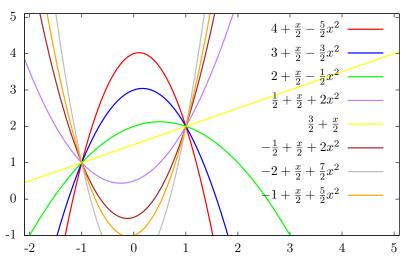
CS 573: Algorithms, Fall 2014

Fast Fourier Transform

Lecture 19 October 30, 2014

Polynomials and point value pairs

Some polynomials of degree two, passing through two fixed points



Multiplying polynomials quickly

Definition

polynomial p(x) of degree n:a function $p(x) = \sum_{j=0}^n a_j x^j = a_0 + x(a_1 + x(a_2 + \ldots + xa_n)).$

 x_0 : $p(x_0)$ can be computed in O(n) time. "dual" (and equivalent) representation...

Theorem

For any set $\left\{(x_0,y_0),(x_1,y_1),\ldots,(x_{n-1},y_{n-1})\right\}$ of n point-value pairs such that all the x_k values are distinct, there is a unique polynomial p(x) of degree n-1, such that $y_k=p(x_k)$, for $k=0,\ldots,n-1$.

Polynomial via point-value

 $ig\{(x_0,y_0),(x_1,y_1),(x_2,y_2)ig\}$: polynomial through points:

$$p(x) = y_0 \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_0-x_0)(x_0-x_1)(x_0-x_2)} \ + y_1 \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_1-x_0)(x_1-x_1)(x_1-x_2)} \ + y_2 \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_2-x_0)(x_2-x_1)(x_2-x_2)}$$

Sariel (UIUC) CS573 Fall 2014

Polynomial via point-value

 $ig\{(x_0,y_0),(x_1,y_1),(x_2,y_2)ig\}$: polynomial through points:

$$egin{split} p(x) &= y_0 rac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} \ &+ y_1 rac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} \ &+ y_2 rac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \end{split}$$

Sariel (UIUC) CS573 4 Fall 2014

Polynomial via point-value

 $\Big\{(x_0,y_0),(x_1,y_1),\ldots,(x_{n-1},y_{n-1})\Big\}$: polynomial through points:

$$p(x) = \sum\limits_{i=0}^{n-1} y_i rac{\prod_{j
eq i} (x-x_j)}{\prod_{j
eq i} (x_i-x_j)}.$$

ith is zero for $x=x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_{n-1}$, and is equal to y_i for $x=x_i$.

Just because.

- lacksquare Given n point-value pairs. Can compute p(x) in $O(n^2)$ time.
- Point-value pairs representation: Multiply polynomials quickly!
- $oldsymbol{0}$ $oldsymbol{p}$, $oldsymbol{q}$ polynomial of degree $oldsymbol{n}-oldsymbol{1}$, both represented by $oldsymbol{2n}$ point-value pairs

$$\Big\{(x_0,y_0),(x_1,y_1),\ldots,(x_{2n-1},y_{2n-1})\Big\}$$
 for $p(x),$ and $\Big\{(x_0,y_0'),(x_1,y_1'),\ldots,(x_{2n-1},y_{2n-1}')\Big\}$ for $q(x).$

 $\mathbf{r}(x) = p(x)q(x)$: product.

Just because.

- lacktriangledown Given n point-value pairs. Can compute p(x) in $O(n^2)$ time.
- Point-value pairs representation: Multiply polynomials quickly!
- $oldsymbol{0}$ $oldsymbol{p}$, $oldsymbol{q}$ polynomial of degree $oldsymbol{n}-oldsymbol{1}$, both represented by $oldsymbol{2}oldsymbol{n}$ point-value pairs

$$\Big\{(x_0,y_0),(x_1,y_1),\ldots,(x_{2n-1},y_{2n-1})\Big\}$$
 for $p(x),$ and $\Big\{(x_0,y_0'),(x_1,y_1'),\ldots,(x_{2n-1},y_{2n-1}')\Big\}$ for $q(x).$

 $\mathbf{r}(x) = \mathbf{p}(x)\mathbf{q}(x)$: product.

Just because.

- lacksquare Given n point-value pairs. Can compute p(x) in $O(n^2)$ time.
- Point-value pairs representation: Multiply polynomials quickly!
- $oldsymbol{0}$ p, q polynomial of degree n-1, both represented by 2n point-value pairs

$$\Big\{(x_0,y_0),(x_1,y_1),\ldots,(x_{2n-1},y_{2n-1})\Big\}$$
 for $p(x),$ and $\Big\{(x_0,y_0'),(x_1,y_1'),\ldots,(x_{2n-1},y_{2n-1}')\Big\}$ for $q(x).$

r(x) = p(x)q(x): product.

Just because.

- lacksquare Given n point-value pairs. Can compute p(x) in $O(n^2)$ time.
- Point-value pairs representation: Multiply polynomials quickly!
- $oldsymbol{0}$ p, q polynomial of degree n-1, both represented by 2n point-value pairs

$$\Big\{(x_0,y_0),(x_1,y_1),\ldots,(x_{2n-1},y_{2n-1})\Big\}$$
 for $p(x),$ and $\Big\{(x_0,y_0'),(x_1,y_1'),\ldots,(x_{2n-1},y_{2n-1}')\Big\}$ for $q(x).$

r(x) = p(x)q(x): product.

Just because.

lacktriangle In point-value representation representation of r(x) is

$$egin{aligned} \left\{ (x_0, r(x_0)), \dots, (x_{2n-1}, r(x_{2n-1}))
ight\} \ &= \left\{ \left(x_0, p(x_0) q(x_0)
ight), \dots, \left(x_{2n-1}, p(x_{2n-1}) q(x_{2n-1})
ight)
ight\} \ &= \left\{ (x_0, y_0 y_0'), \dots, (x_{2n-1}, y_{2n-1} y_{2n-1}')
ight\}. \end{aligned}$$

- $oldsymbol{0} p(x)$ and q(x): point-value pairs \Longrightarrow compute r(x)=p(x)q(x) in linear time!
- $@ \dots$ but r(x) is in point-value representation. Bummer
- ullet ...but we can compute r(x) from this representation.
- ① Purpose: Translate quickly (i.e., $O(n \log n)$ time) from the standard r to point-value pairs representation of polynomials.
- ...and back!
- \Longrightarrow computing product of two polynomials in $O(n \log n)$ time.
- Fast Fourier Transform is a way to do this.
- ullet choosing the $oldsymbol{x}_i$ values carefully, and using divide and conquer.

- $oldsymbol{0} p(x)$ and q(x): point-value pairs \Longrightarrow compute r(x)=p(x)q(x) in linear time!
- ② ...but r(x) is in point-value representation. Bummer.
- ullet ...but we can compute r(x) from this representation.
- ① Purpose: Translate quickly (i.e., $O(n \log n)$ time) from the standard r to point-value pairs representation of polynomials.
- ...and back!
- \circledcirc \Longrightarrow computing product of two polynomials in $O(n \log n)$ time.
- Fast Fourier Transform is a way to do this.
- ullet choosing the x_i values carefully, and using divide and conquer

- $oldsymbol{0} p(x)$ and q(x): point-value pairs \Longrightarrow compute r(x)=p(x)q(x) in linear time!
- ② ...but r(x) is in point-value representation. Bummer.
- lacksquare ...but we can compute r(x) from this representation.
- ① Purpose: Translate quickly (i.e., $O(n \log n)$ time) from the standard r to point-value pairs representation of polynomials.
- ...and back!
- \circledcirc \Longrightarrow computing product of two polynomials in $O(n \log n)$ time.
- Fast Fourier Transform is a way to do this.
- ullet choosing the x_i values carefully, and using divide and conquer.

- $oldsymbol{0} p(x)$ and q(x): point-value pairs \Longrightarrow compute r(x)=p(x)q(x) in linear time!
- ② ...but r(x) is in point-value representation. Bummer.
- lacksquare ...but we can compute r(x) from this representation.
- **9** Purpose: Translate quickly (i.e., $O(n \log n)$ time) from the standard r to point-value pairs representation of polynomials.
- ...and back!
- \Longrightarrow computing product of two polynomials in $O(n \log n)$ time.
- Fast Fourier Transform is a way to do this.
- ullet choosing the x_i values carefully, and using divide and conquer.

- $oldsymbol{0} p(x)$ and q(x): point-value pairs \implies compute r(x) = p(x)q(x) in linear time!
- ② ...but r(x) is in point-value representation. Bummer.
- lacksquare ...but we can compute r(x) from this representation.
- **9** Purpose: Translate quickly (i.e., $O(n \log n)$ time) from the standard r to point-value pairs representation of polynomials.
- ...and back!
- \Longrightarrow computing product of two polynomials in $O(n \log n)$ time.
- Fast Fourier Transform is a way to do this.
- ullet choosing the x_i values carefully, and using divide and conquer

- $oldsymbol{0} p(x)$ and q(x): point-value pairs \Longrightarrow compute r(x)=p(x)q(x) in linear time!
- ② ...but r(x) is in point-value representation. Bummer.
- lacksquare ...but we can compute r(x) from this representation.
- **9** Purpose: Translate quickly (i.e., $O(n \log n)$ time) from the standard r to point-value pairs representation of polynomials.
- ...and back!
- \Longrightarrow computing product of two polynomials in $O(n \log n)$ time.
- Fast Fourier Transform is a way to do this.
- ${ t @}$ choosing the x_i values carefully, and using divide and conquer.

- $oldsymbol{0} p(x)$ and q(x): point-value pairs \implies compute r(x) = p(x)q(x) in linear time!
- ② ...but r(x) is in point-value representation. Bummer.
- lacksquare ...but we can compute r(x) from this representation.
- **9** Purpose: Translate quickly (i.e., $O(n \log n)$ time) from the standard r to point-value pairs representation of polynomials.
- ...and back!
- \Longrightarrow computing product of two polynomials in $O(n \log n)$ time.
- **Fast Fourier Transform** is a way to do this.
- ullet choosing the x_i values carefully, and using divide and conquer.

- $lackbox{0} p(x)$ and q(x): point-value pairs \implies compute r(x)=p(x)q(x) in linear time!
- ② ...but r(x) is in point-value representation. Bummer.
- lacksquare ...but we can compute r(x) from this representation.
- **9** Purpose: Translate quickly (i.e., $O(n \log n)$ time) from the standard r to point-value pairs representation of polynomials.
- ...and back!
- \Longrightarrow computing product of two polynomials in $O(n \log n)$ time.
- **7** Fast Fourier Transform is a way to do this.
- lacktriangle choosing the x_i values carefully, and using divide and conquer.

Part I

Computing a polynomial quickly on n values

- **1** Assume: polynomials have degree n-1, where $n=2^k$.
- ② .. pad polynomials with terms having zero coefficients.
- **1** Magic set of numbers: $\Psi=\{x_1,\ldots,x_n\}$. Property: $|\mathsf{SQ}(\Psi)|=n/2$, where $\mathsf{SQ}(\Psi)=\left\{x^2\ \middle|\ x\in\Psi\right\}$.
- Easy to find such set...
- Magic: Have this property repeatedly... $SQ(SQ(\Psi))$ has n/4 distinct values.
- ${\mathbb O}$ ${\mathsf {SQ}}({\mathsf {SQ}}({\mathsf {SQ}}(\Psi)))$ has n/8 values.
- \bigcirc $\mathsf{SQ}^i(\Psi)$ has $n/2^i$ distinct values.
- Oops: No such set of real numbers.
- NO SUCH SET.

- **1** Assume: polynomials have degree n-1, where $n=2^k$.
- .. pad polynomials with terms having zero coefficients.
- $\hbox{\bf 0} \ \, \textit{Magic set} \ \, \text{of numbers:} \ \, \Psi = \{x_1, \dots, x_n\}. \\ \hbox{Property:} \ \, |\mathsf{SQ}(\Psi)| = n/2, \ \, \text{where} \ \, \mathsf{SQ}(\Psi) = \left\{x^2 \ \middle| \ \, x \in \Psi\right\}.$
- Easy to find such set...
- Magic: Have this property repeatedly... $\mathsf{SQ}(\mathsf{SQ}(\Psi))$ has n/4 distinct values.
- **O** $\mathsf{SQ}(\mathsf{SQ}(\mathsf{SQ}(\Psi)))$ has n/8 values.
- \bigcirc $\mathsf{SQ}^i(\Psi)$ has $n/2^i$ distinct values.
- Oops: No such set of real numbers.
- NO SUCH SET.

- **1** Assume: polynomials have degree n-1, where $n=2^k$.
- .. pad polynomials with terms having zero coefficients.
- **1** Magic set of numbers: $\Psi=\{x_1,\ldots,x_n\}$. Property: $|\mathsf{SQ}(\Psi)|=n/2$, where $\mathsf{SQ}(\Psi)=\{x^2\mid x\in\Psi\}$.
- Easy to find such set...
- Magic: Have this property repeatedly... $\mathsf{SQ}(\mathsf{SQ}(\Psi))$ has n/4 distinct values.
- ${\mathbb O}$ ${\mathsf {SQ}}({\mathsf {SQ}}({\mathsf {SQ}}(\Psi)))$ has n/8 values.
- \bigcirc **SQ**ⁱ(Ψ) has $n/2^i$ distinct values.
- Oops: No such set of real numbers.
- NO SUCH SET.

- **1** Assume: polynomials have degree n-1, where $n=2^k$.
- .. pad polynomials with terms having zero coefficients.
- **3** *Magic set* of numbers: $\Psi=\{x_1,\ldots,x_n\}$. Property: $|\mathsf{SQ}(\Psi)|=n/2$, where $\mathsf{SQ}(\Psi)=\left\{x^2\ \middle|\ x\in\Psi\right\}$.
- Easy to find such set...
- Magic: Have this property repeatedly... $\mathsf{SQ}(\mathsf{SQ}(\Psi))$ has n/4 distinct values.
- ${\mathbb O}$ ${\mathsf {SQ}}({\mathsf {SQ}}({\mathsf {SQ}}(\Psi)))$ has n/8 values.
- **3** $\mathbf{SQ}^i(\Psi)$ has $n/2^i$ distinct values.
- Oops: No such set of real numbers.
- NO SUCH SET.

- **1** Assume: polynomials have degree n-1, where $n=2^k$.
- .. pad polynomials with terms having zero coefficients.
- **3** *Magic set* of numbers: $\Psi=\{x_1,\ldots,x_n\}$. Property: $|\mathsf{SQ}(\Psi)|=n/2$, where $\mathsf{SQ}(\Psi)=\left\{x^2\ \middle|\ x\in\Psi\right\}$.
- Easy to find such set...
- Magic: Have this property repeatedly... $SQ(SQ(\Psi))$ has n/4 distinct values.
- **O** $\mathsf{SQ}(\mathsf{SQ}(\mathsf{SQ}(\Psi)))$ has n/8 values.
- \bigcirc $\mathsf{SQ}^i(\Psi)$ has $n/2^i$ distinct values.
- Oops: No such set of real numbers.
- NO SUCH SET.

- **1** Assume: polynomials have degree n-1, where $n=2^k$.
- .. pad polynomials with terms having zero coefficients.
- **3** *Magic set* of numbers: $\Psi=\{x_1,\ldots,x_n\}$. Property: $|\mathsf{SQ}(\Psi)|=n/2$, where $\mathsf{SQ}(\Psi)=\left\{x^2\ \middle|\ x\in\Psi\right\}$.
- $|\text{square}(\Psi)| = |\Psi|/2.$
- Easy to find such set...
- Magic: Have this property repeatedly... $\mathsf{SQ}(\mathsf{SQ}(\Psi))$ has n/4 distinct values.
- ${\mathbb O}$ ${\mathsf {SQ}}({\mathsf {SQ}}({\mathsf {SQ}}(\Psi)))$ has n/8 values.
- \bigcirc $\mathsf{SQ}^i(\Psi)$ has $n/2^i$ distinct values.
- Oops: No such set of real numbers.
- NO SUCH SET.

- **1** Assume: polynomials have degree n-1, where $n=2^k$.
- .. pad polynomials with terms having zero coefficients.
- **3** *Magic set* of numbers: $\Psi=\{x_1,\ldots,x_n\}$. Property: $|\mathsf{SQ}(\Psi)|=n/2$, where $\mathsf{SQ}(\Psi)=\left\{x^2\ \middle|\ x\in\Psi\right\}$.
- $|\text{square}(\Psi)| = |\Psi|/2.$
- Easy to find such set...
- Magic: Have this property repeatedly... $SQ(SQ(\Psi))$ has n/4 distinct values.
- ${\mathbb O}$ SQ(SQ(SQ(Ψ))) has n/8 values.
- \bigcirc $\mathsf{SQ}^i(\Psi)$ has $n/2^i$ distinct values.
- Oops: No such set of real numbers.
 - NO SUCH SET.

- **1** Assume: polynomials have degree n-1, where $n=2^k$.
- .. pad polynomials with terms having zero coefficients.
- **3** *Magic set* of numbers: $\Psi=\{x_1,\ldots,x_n\}$. Property: $|\mathsf{SQ}(\Psi)|=n/2$, where $\mathsf{SQ}(\Psi)=\left\{x^2\ \middle|\ x\in\Psi\right\}$.
- $|\text{square}(\Psi)| = |\Psi|/2.$
- Easy to find such set...
- Magic: Have this property repeatedly... $SQ(SQ(\Psi))$ has n/4 distinct values.
- ${f O}$ ${f SQ}({f SQ}({f SQ}(\Psi)))$ has n/8 values.
- \bigcirc $\mathsf{SQ}^i(\Psi)$ has $n/2^i$ distinct values.
- Oops: No such set of real numbers.
 - NO SUCH SET.

- **1** Assume: polynomials have degree n-1, where $n=2^k$.
- .. pad polynomials with terms having zero coefficients.
- **3** *Magic set* of numbers: $\Psi=\{x_1,\ldots,x_n\}$. Property: $|\mathsf{SQ}(\Psi)|=n/2$, where $\mathsf{SQ}(\Psi)=\left\{x^2\ \middle|\ x\in\Psi\right\}$.
- $|\text{square}(\Psi)| = |\Psi|/2.$
- Easy to find such set...
- Magic: Have this property repeatedly... $SQ(SQ(\Psi))$ has n/4 distinct values.
- ${f O}$ ${f SQ}({f SQ}({f SQ}(\Psi)))$ has n/8 values.
- **3** $\mathbf{SQ}^i(\Psi)$ has $n/2^i$ distinct values.
- Oops: No such set of real numbers.
- NO SUCH SET.

- **1** Assume: polynomials have degree n-1, where $n=2^k$.
- .. pad polynomials with terms having zero coefficients.
- **3** *Magic set* of numbers: $\Psi=\{x_1,\ldots,x_n\}$. Property: $|\mathsf{SQ}(\Psi)|=n/2$, where $\mathsf{SQ}(\Psi)=\left\{x^2\ \middle|\ x\in\Psi\right\}$.
- $|\text{square}(\Psi)| = |\Psi|/2.$
- Easy to find such set...
- Magic: Have this property repeatedly... $SQ(SQ(\Psi))$ has n/4 distinct values.
- **O** $\mathsf{SQ}(\mathsf{SQ}(\mathsf{SQ}(\Psi)))$ has n/8 values.
- **3** $\mathbf{SQ}^i(\Psi)$ has $n/2^i$ distinct values.
- Oops: No such set of real numbers.
- NO SUCH SET.

- **1** Assume: polynomials have degree n-1, where $n=2^k$.
- .. pad polynomials with terms having zero coefficients.
- **10** *Magic set* of numbers: $\Psi = \{x_1, \ldots, x_n\}$. Property: $|\mathsf{SQ}(\Psi)| = n/2$, where $\mathsf{SQ}(\Psi) = \{x^2 \mid x \in \Psi\}$.
- Easy to find such set...
- Magic: Have this property repeatedly... $SQ(SQ(\Psi))$ has n/4 distinct values.
- ${f O}$ ${f SQ}({f SQ}({f SQ}(\Psi)))$ has n/8 values.
- **3** $\mathbf{SQ}^i(\Psi)$ has $n/2^i$ distinct values.
- Oops: No such set of real numbers.
- NO SUCH SET.

Collapsible sets

Assume magic...

Let us for the time being ignore this technicality, and fly, for a moment, into the land of fantasy, and assume that we do have such a set of numbers, so that $|\mathbf{SQ}^i(\Psi)| = n/2^i$ numbers, for $i=0,\ldots,k$. Let us call such a set of numbers *collapsible*.

... two polynomials of half the degree

lacksquare For a set $\mathfrak{X} = \{x_0, \dots, x_n\}$ and polynomial p(x), let

$$pig(\mathfrak{X}ig) = \left\langle \left(x_0, p(x_0)
ight), \ldots, \left(x_n, p(x_n)
ight)
ight
angle.$$

$$u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$$
 and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i$.

- 3 all even degree terms in $u(\cdot)$, all odd degree terms in $v(\cdot)$.
- 1 maximum degree of u(y), v(y) is n/2.

... two polynomials of half the degree

lacktriangledown For a set $\mathfrak{X}=\{x_0,\ldots,x_n\}$ and polynomial p(x), let

$$pig(\mathfrak{X}ig) = \left\langle \left(x_0, p(x_0)
ight), \ldots, \left(x_n, p(x_n)
ight)
ight
angle.$$

$$u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$$
 and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i$.

- 3 all even degree terms in $u(\cdot)$, all odd degree terms in $v(\cdot)$.
- 1 maximum degree of u(y), v(y) is n/2.

... two polynomials of half the degree

lacksquare For a set $\mathfrak{X} = \{x_0, \dots, x_n\}$ and polynomial p(x), let

$$pig(\mathfrak{X}ig) = \left\langle \left(x_0, p(x_0)
ight), \ldots, \left(x_n, p(x_n)
ight)
ight
angle.$$

$$u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$$
 and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i.$

- 3 all even degree terms in $u(\cdot)$, all odd degree terms in $v(\cdot)$.
- 1 maximum degree of u(y), v(y) is n/2.

... two polynomials of half the degree

lacksquare For a set $\mathfrak{X}=\{x_0,\ldots,x_n\}$ and polynomial p(x), let

$$pig(\mathfrak{X}ig) = \left\langle \left(x_0, p(x_0)
ight), \ldots, \left(x_n, p(x_n)
ight)
ight
angle.$$

$$u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$$
 and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i$.

- 3 all even degree terms in $u(\cdot)$, all odd degree terms in $v(\cdot)$.
- lacktriangledown maximum degree of u(y), v(y) is n/2.

- $lacksquare p(x) = \sum_{i=0}^{n-1} a_i x^i$ as $p(x) = u(x^2) + x \cdot v(x^2)$.
- ② Ψ : collapsible set of size n.
- 0 $p(\Psi)$: compute polynomial of degree n-1 on n values.
- Decompose

$$u(y)=\sum\limits_{i=0}^{n/2-1}a_{2i}y^i$$
 and $v(y)=\sum\limits_{i=0}^{n/2-1}a_{1+2i}y^i.$

- lacksquare Need to compute $v(x^2)$, for all $x\in\Psi$.
- \blacksquare Need to compute $u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$.
- $oxed{0} \ u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: comp. poly. degree $rac{n}{2}-1$ on $rac{n}{2}$ values.

- **2** Ψ : collapsible set of size n.
- 0 $p(\Psi)$: compute polynomial of degree n-1 on n values.
- Decompose:

$$u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$$
 and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i.$

- lacksquare Need to compute $v(x^2)$, for all $x\in\Psi$.
- \blacksquare Need to compute $u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$.
- $oxed{0} \ u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: comp. poly. degree $rac{n}{2}-1$ on $rac{n}{2}$ values.

- $lacksquare p(x) = \sum_{i=0}^{n-1} a_i x^i$ as $p(x) = u(x^2) + x \cdot v(x^2)$.
- **2** Ψ : collapsible set of size n.
- **3** $p(\Psi)$: compute polynomial of degree n-1 on n values.
- Decompose:

$$u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$$
 and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i.$

- lacktriangledown Need to compute $v(x^2)$, for all $x\in\Psi$.
- \blacksquare Need to compute $u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$.
- $ext{ 0 } u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: comp. poly. degree $rac{n}{2}-1$ on $rac{n}{2}$ values.

- $\mathbf{2} \quad \Psi$: collapsible set of size n.
- **1** $p(\Psi)$: compute polynomial of degree n-1 on n values.
- Decompose:

$$u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$$
 and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i.$

- **1** Need to compute $v(x^2)$, for all $x \in \Psi$.
- \blacksquare Need to compute $u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$.
- $oldsymbol{0} \ u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: comp. poly. degree $rac{n}{2}-1$ on $rac{n}{2}$ values.

- $lackbox{0} p(x) = \sum_{i=0}^{n-1} a_i x^i ext{ as } p(x) = u(x^2) + x \cdot v(x^2).$
- $\mathbf{2} \quad \Psi$: collapsible set of size n.
- **1** $p(\Psi)$: compute polynomial of degree n-1 on n values.
- Decompose:

$$u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$$
 and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i.$

- **1** Need to compute $u(x^2)$, for all $x \in \Psi$.
- **1** Need to compute $v(x^2)$, for all $x \in \Psi$.
- \bullet Need to compute $u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$.
- $oldsymbol{0} \ u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: comp. poly. degree $rac{n}{2}-1$ on $rac{n}{2}$ values.

- $lackbox{0} p(x) = \sum_{i=0}^{n-1} a_i x^i$ as $p(x) = u(x^2) + x \cdot v(x^2)$.
- **2** Ψ : collapsible set of size n.
- **1** $p(\Psi)$: compute polynomial of degree n-1 on n values.
- Decompose:

$$u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$$
 and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i.$

- **1** Need to compute $v(x^2)$, for all $x \in \Psi$.
- $\qquad \Longrightarrow \ \mathsf{Need to \ compute} \ u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi)).$
- $oldsymbol{0} \ u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: comp. poly. degree $rac{n}{2}-1$ on $rac{n}{2}$ values.

- $lackbox{0} p(x) = \sum_{i=0}^{n-1} a_i x^i$ as $p(x) = u(x^2) + x \cdot v(x^2)$.
- **2** Ψ : collapsible set of size n.
- **1** $p(\Psi)$: compute polynomial of degree n-1 on n values.
- Decompose:

$$u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$$
 and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i.$

- **1** Need to compute $u(x^2)$, for all $x \in \Psi$.
- **6** Need to compute $v(x^2)$, for all $x \in \Psi$.
- $lackbox{O} \mathsf{SQ}(\Psi) = ig\{ x^2 \ ig| \ x \in \Psi ig\}.$
- \blacksquare Need to compute $u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$.
- $oldsymbol{0} \ u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: comp. poly. degree $rac{n}{2}-1$ on $rac{n}{2}$ values.

- $\ \ \ \ \Psi$: collapsible set of size n.
- **1** $p(\Psi)$: compute polynomial of degree n-1 on n values.
- Decompose:

$$u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$$
 and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i.$

- **1** Need to compute $u(x^2)$, for all $x \in \Psi$.
- **1** Need to compute $v(x^2)$, for all $x \in \Psi$.
- $lackbox{O} \mathsf{SQ}(\Psi) = ig\{ x^2 \ ig| \ x \in \Psi ig\}.$
- lacksquare Need to compute $u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$.
- $extbf{0} \ u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: comp. poly. degree $rac{n}{2}-1$ on $rac{n}{2}$ values.

- $oldsymbol{0} p(x) = \sum_{i=0}^{n-1} a_i x^i$ as $p(x) = u(x^2) + x \cdot v(x^2)$.
- **2** Ψ : collapsible set of size n.
- **1** $p(\Psi)$: compute polynomial of degree n-1 on n values.
- Oecompose:

$$u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$$
 and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i.$

- **1** Need to compute $u(x^2)$, for all $x \in \Psi$.
- **1** Need to compute $v(x^2)$, for all $x \in \Psi$.
- $lackbox{O} \mathsf{SQ}(\Psi) = ig\{ x^2 \ ig| \ x \in \Psi ig\}.$
- \bullet Need to compute $u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$.
- $oldsymbol{0} \ u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: comp. poly. degree $rac{n}{2}-1$ on $rac{n}{2}$ values.

- **1** Ψ : Collapsible set of size n.
- ① $p(x) = \sum_{i=0}^{n-1} a_i x^i$ as $p(x) = u(x^2) + x \cdot v(x^2)$.
- $u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$ and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i$.
- ${\color{red} \bullet} \ u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: Computed recursively.
- lacksquare Need to compute $p(\Psi)$
- $lackbox{0}$ For $x\in\Psi$: Compute $p(x)=u(x^2)+x\cdot v(x^2)$.
- extstyle ext
- \odot Takes O(n) time overall.

- **1** Ψ : Collapsible set of size n.
- ② $p(x) = \sum_{i=0}^{n-1} a_i x^i$ as $p(x) = u(x^2) + x \cdot v(x^2)$.
- $0 \ u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$ and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i$.
- $u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: Computed recursively.
- lacksquare Need to compute $p(\Psi)$.
- \bullet For $x \in \Psi$: Compute $p(x) = u(x^2) + x \cdot v(x^2)$.
- $extbf{ iny Takes}$ constant time per single element $x\in\Psi.$
- 1 Takes O(n) time overall.

- **1** Ψ : Collapsible set of size n.
- ② $p(x) = \sum_{i=0}^{n-1} a_i x^i$ as $p(x) = u(x^2) + x \cdot v(x^2)$.
- $lacksquare u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$ and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i.$
- $u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: Computed recursively.
- lacksquare Need to compute $p(\Psi)$.
- $lackbox{0}$ For $x\in\Psi$: Compute $p(x)=u(x^2)+x\cdot v(x^2)$.
- $extbf{ iny Takes}$ constant time per single element $x \in \Psi$.
- \odot Takes O(n) time overall.

- Ψ : Collapsible set of size n.
- ② $p(x) = \sum_{i=0}^{n-1} a_i x^i$ as $p(x) = u(x^2) + x \cdot v(x^2)$.
- $ullet u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$ and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i.$
- **1** $u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: Computed recursively.
- ullet Need to compute $p(\Psi)$.
- $lackbox{0}$ For $x \in \Psi$: Compute $p(x) = u(x^2) + x \cdot v(x^2)$.
- $extbf{ iny Takes}$ constant time per single element $x \in \Psi$.
- \odot Takes O(n) time overall.

- Ψ : Collapsible set of size n.
- ② $p(x) = \sum_{i=0}^{n-1} a_i x^i$ as $p(x) = u(x^2) + x \cdot v(x^2)$.
- $ullet u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$ and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i.$
- **1** $u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: Computed recursively.
- **1** Need to compute $p(\Psi)$.
- $lackbox{0}$ For $x\in\Psi$: Compute $p(x)=u(x^2)+x\cdot v(x^2)$.
- $extbf{ iny Takes}$ constant time per single element $x \in \Psi$.
- \odot Takes O(n) time overall.

- Ψ : Collapsible set of size n.
- ② $p(x) = \sum_{i=0}^{n-1} a_i x^i$ as $p(x) = u(x^2) + x \cdot v(x^2)$.
- $lacksquare u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$ and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i.$
- $u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: Computed recursively.
- **1** Need to compute $p(\Psi)$.
- $extbf{ iny Takes}$ constant time per single element $x \in \Psi$
- \odot Takes O(n) time overall.

- Ψ : Collapsible set of size n.
- ② $p(x) = \sum_{i=0}^{n-1} a_i x^i$ as $p(x) = u(x^2) + x \cdot v(x^2)$.
- $lacksquare u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$ and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i.$
- $u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: Computed recursively.
- **1** Need to compute $p(\Psi)$.
- $lackbox{0}$ For $x \in \Psi$: Compute $p(x) = u(x^2) + x \cdot v(x^2)$.
- lacksquare Takes constant time per single element $x\in\Psi$.
- ${\color{red} {f 0}}$ Takes O(n) time overall.

- \bullet Ψ : Collapsible set of size n.
- $p(x) = \sum_{i=0}^{n-1} a_i x^i$ as $p(x) = u(x^2) + x \cdot v(x^2)$.
- $ullet u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i$ and $v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i$.
- **4** $u(\mathsf{SQ}(\Psi)), v(\mathsf{SQ}(\Psi))$: Computed recursively.
- **1** Need to compute $p(\Psi)$.
- **6** For $x \in \Psi$: Compute $p(x) = u(x^2) + x \cdot v(x^2)$.
- **1** Takes constant time per single element $x \in \Psi$.
- **3** Takes O(n) time overall.

FFT algorithm

```
\mathsf{FFTAlg}(p, X)
           input: p(x): A polynomial of degree n: p(x) = \sum_{i=0}^{n-1} a_i x^i
                            X: A collapsible set of n elements.
           output: p(X)
     u(y) = \sum_{i=0}^{n/2-1} a_{2i} y^i

v(y) = \sum_{i=0}^{n/2-1} a_{1+2i} y^i.
     Y = \mathsf{SQ}(X) = \left\{ x^2 \mid x \in X \right\}.
      U = \mathsf{FFTAlg}(u, Y) /* U = u(Y) */
      V = \mathsf{FFTAlg}(v, Y) /* V = v(Y) */
      Out \leftarrow \emptyset
     for x \in X do
           /* p(x) = u(x^2) + x * v(x^2), U[x^2] is the value u(x^2) * / (x^2) = u(x^2) + x * v(x^2)
           (x, p(x)) \leftarrow (x, U[x^2] + x \cdot V[x^2])
           Out \leftarrow Out \cup \{(x, p(x))\}
```

return Out

Running time analysis...

...an old foe emerges once again to serve

- T(m, n): Time of computing a polynomial of degree m on n values.
- We have that:

$$T(n-1,n) = 2T(n/2-1,n/2) + O(n).$$

① The solution to this recurrence is $O(n \log n)$.

Running time analysis...

...an old foe emerges once again to serve

- T(m, n): Time of computing a polynomial of degree m on n values.
- We have that:

$$T(n-1,n) = 2T(n/2-1,n/2) + O(n).$$

1 The solution to this recurrence is $O(n \log n)$.

Running time analysis...

...an old foe emerges once again to serve

- T(m, n): Time of computing a polynomial of degree m on n values.
- We have that:

$$T(n-1,n) = 2T(n/2-1,n/2) + O(n).$$

3 The solution to this recurrence is $O(n \log n)$.

Generating Collapsible Sets

• How to generate collapsible sets?

Generating Collapsible Sets

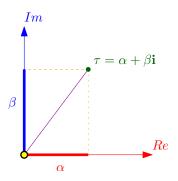
- How to generate collapsible sets?
- Trick: Use complex numbers!

- Complex number: pair (α, β) of real numbers. Written as $\tau = \alpha + i\beta$.
- ② α : **real** part, β : **imaginary** part.
- \circ i is the root of -1.
- Geometrically: a point in the complex plane:
- ① polar form: $\tau = r \cos \phi + \mathrm{i} r \sin \phi = r(\cos \phi + \mathrm{i} \sin \phi)$
- ② $r = \sqrt{\alpha^2 + \beta^2}$ and $\phi = \arcsin(\beta/\alpha)$.

- Complex number: pair (α, β) of real numbers. Written as $\tau = \alpha + i\beta$.
- ② α : **real** part, β : **imaginary** part.
- \odot i is the root of -1.
- Geometrically: a point in the complex plane:
- ① polar form: $\tau = r \cos \phi + ir \sin \phi = r(\cos \phi + i \sin \phi)$
- ② $r = \sqrt{\alpha^2 + \beta^2}$ and $\phi = \arcsin(\beta/\alpha)$.

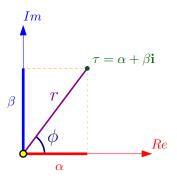
- Complex number: pair (α, β) of real numbers. Written as $\tau = \alpha + i\beta$.
- ② α : **real** part, β : **imaginary** part.
- 0 i is the root of -1.
- Geometrically: a point in the complex plane:
- ① polar form: $\tau = r\cos\phi + \mathrm{i}r\sin\phi = r(\cos\phi + \mathrm{i}\sin\phi)$
- ② $r = \sqrt{\alpha^2 + \beta^2}$ and $\phi = \arcsin(\beta/\alpha)$.

- Complex number: pair (α, β) of real numbers. Written as $\tau = \alpha + i\beta$.
- ② α : **real** part, β : **imaginary** part.
- $\mathbf{0}$ **i** is the root of $\mathbf{-1}$.
- Geometrically: a point in the complex plane:



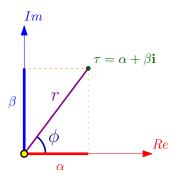
- ① polar form: $\tau = r \cos \phi + ir \sin \phi = r(\cos \phi + i \sin \phi)$
- $r = \sqrt{\alpha^2 + \beta^2}$ and $\phi = \arcsin(\beta/\alpha)$.

- Complex number: pair (α, β) of real numbers. Written as $\tau = \alpha + i\beta$.
- ② α : **real** part, β : **imaginary** part.
- $\mathbf{0}$ **i** is the root of $\mathbf{-1}$.
- Geometrically: a point in the complex plane:



- **1** polar form: $au = r\cos\phi + \mathrm{i}r\sin\phi = r(\cos\phi + \mathrm{i}\sin\phi)$
- $r = \sqrt{\alpha^2 + \beta^2}$ and $\phi = \arcsin(\beta/\alpha)$.

- Complex number: pair (α, β) of real numbers. Written as $\tau = \alpha + i\beta$.
- ② α : **real** part, β : **imaginary** part.
- $\mathbf{3}$ **i** is the root of $\mathbf{-1}$.
- Geometrically: a point in the complex plane:



- **1** polar form: $\tau = r\cos\phi + \mathrm{i}r\sin\phi = r(\cos\phi + \mathrm{i}\sin\phi)$
- $r = \sqrt{\alpha^2 + \beta^2}$ and $\phi = \arcsin(\beta/\alpha)$.

A useful formula: $\cos\phi + \mathrm{i}\sin\phi = \mathrm{e}^{\mathrm{i}\phi}$

By Taylor's expansion:

$$\sin x = x - rac{x^3}{3!} + rac{x^5}{5!} - rac{x^7}{7!} + \cdots \; ,$$
 $\cos x = 1 - rac{x^2}{2!} + rac{x^4}{4!} - rac{x^6}{6!} + \cdots \; ,$ and $e^x = 1 + rac{x}{1!} + rac{x^2}{2!} + rac{x^3}{3!} + \cdots \; .$

$$e^{\mathrm{i}x} = 1 + \mathrm{i}rac{x}{1!} - rac{x^2}{2!} - \mathrm{i}rac{x^3}{3!} + rac{x^4}{4!} + \mathrm{i}rac{x^5}{5!} - rac{x^6}{6!} \cdots \ = \cos x + \mathrm{i}\sin x.$$

A useful formula: $\cos\phi + \mathrm{i}\sin\phi = \mathrm{e}^{\mathrm{i}\phi}$

By Taylor's expansion:

$$\sin x = x - rac{x^3}{3!} + rac{x^5}{5!} - rac{x^7}{7!} + \cdots \; ,$$
 $\cos x = 1 - rac{x^2}{2!} + rac{x^4}{4!} - rac{x^6}{6!} + \cdots \; ,$ and $e^x = 1 + rac{x}{1!} + rac{x^2}{2!} + rac{x^3}{3!} + \cdots \; .$

② Since $i^2 = -1$:

$$e^{ix} = 1 + i\frac{x}{1!} - \frac{x^2}{2!} - i\frac{x^3}{3!} + \frac{x^4}{4!} + i\frac{x^5}{5!} - \frac{x^6}{6!} \cdots$$

= $\cos x + i\sin x$.

polar form:

$$au=r\cos\phi+{
m i}r\sin\phi=r(\cos\phi+{
m i}\sin\phi)=re^{i\phi}$$
 ,

- ② $au=re^{\mathrm{i}\phi}$, $au'=r'e^{\mathrm{i}\phi'}$: complex numbers.
- $au \cdot au' = r e^{\mathrm{i}\phi} \cdot r' e^{\mathrm{i}\phi'} = r r' e^{\mathrm{i}(\phi + \phi')}$
- \bullet $e^{\mathrm{i}\phi}$ is 2π periodic (i.e., $e^{\mathrm{i}\phi}=e^{\mathrm{i}(\phi+2\pi)}$), and $1=e^{\mathrm{i}0}$.
- $oldsymbol{0} au = r e^{\mathrm{i}\phi}$, such that $oldsymbol{ au}^n = r^n e^{\mathrm{i}n\phi} = e^{\mathrm{i}0}$.
- $extbf{0} \implies r=1$, and there must be an integer j, such that

$$n\phi = 0 + 2\pi j \implies \phi = j(2\pi/n).$$

Sariel (UIUC) CS573 20 Fall 2014 20 / 40

- polar form:
 - $au = r\cos\phi + \mathrm{i}r\sin\phi = r(\cos\phi + \mathrm{i}\sin\phi) = re^{i\phi}$,
- $extbf{2} au = r e^{\mathrm{i}\phi}, \quad au' = r' e^{\mathrm{i}\phi'}$: complex numbers.
- $\mathbf{3} \ \mathbf{\tau} \cdot \mathbf{\tau}' = r e^{\mathrm{i}\phi} \cdot r' e^{\mathrm{i}\phi'} = r r' e^{\mathrm{i}(\phi + \phi')}$
- \bullet $e^{\mathrm{i}\phi}$ is 2π periodic (i.e., $e^{\mathrm{i}\phi}=e^{\mathrm{i}(\phi+2\pi)}$), and $1=e^{\mathrm{i}0}$.
- \bullet $au=re^{\mathrm{i}\phi}$, such that $au^n=r^ne^{\mathrm{i}n\phi}=e^{\mathrm{i}0}$.
- $extbf{0} \implies r=1$, and there must be an integer j, such that

$$n\phi = 0 + 2\pi j \implies \phi = j(2\pi/n).$$

Sariel (UIUC) CS573 20 Fall 2014 20 / 40

- polar form:
 - $au = r\cos\phi + \mathrm{i}r\sin\phi = r(\cos\phi + \mathrm{i}\sin\phi) = re^{i\phi}$,
- $oldsymbol{arphi} au = r e^{\mathrm{i}\phi}$, $au' = r' e^{\mathrm{i}\phi'}$: complex numbers.
- \bullet $e^{\mathrm{i}\phi}$ is 2π periodic (i.e., $e^{\mathrm{i}\phi}=e^{\mathrm{i}(\phi+2\pi)}$), and $1=e^{\mathrm{i}0}$.
- \bullet $au=re^{\mathrm{i}\phi}$, such that $au^n=r^ne^{\mathrm{i}n\phi}=e^{\mathrm{i}0}$.
- $extbf{0} \implies r=1$, and there must be an integer j, such that

$$n\phi = 0 + 2\pi j \implies \phi = j(2\pi/n).$$

Sariel (UIUC) CS573 20 Fall 2014 20 / 40

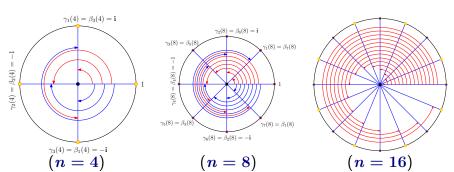
- polar form:
 - $au = r\cos\phi + \mathrm{i}r\sin\phi = r(\cos\phi + \mathrm{i}\sin\phi) = re^{i\phi}$
- $extbf{2} au = r e^{\mathrm{i}\phi}$, $au' = r' e^{\mathrm{i}\phi'}$: complex numbers.
- ullet $e^{\mathrm{i}\phi}$ is 2π periodic (i.e., $e^{\mathrm{i}\phi}=e^{\mathrm{i}(\phi+2\pi)}$), and $1=e^{\mathrm{i}0}$.
- **1** In the following n of n: complex number n raise it to power n get n.
- $oldsymbol{0} au = r e^{\mathrm{i}\phi}$, such that $au^n = r^n e^{\mathrm{i}n\phi} = e^{\mathrm{i}0}$.
- $oldsymbol{0} \implies r=1$, and there must be an integer $oldsymbol{j}$, such that

$$n\phi=0+2\pi j \implies \phi=j(2\pi/n).$$

Roots of unity

The desire to avoid war?

For $j = 0, \dots, n-1$, we get the n distinct **roots of unity**.



- lacksquare Can do all basic calculations on complex numbers in O(1) time.
- Idea: Work over the complex numbers
- Use roots of unity!
- $\ \ \, \gamma\colon n$ th root of unity. There are n such roots, and let $\gamma_j(n)$ denote the jth root.

$$\gamma_j(n) = \cos((2\pi j)/n) + \mathrm{i}\sin((2\pi j)/n) = \gamma^j.$$

Let
$$\mathcal{A}(n) = \{\gamma_0(n), \ldots, \gamma_{n-1}(n)\}.$$

- $|\mathbf{SQ}(\mathcal{A}(n))|$ has n/2 entries.
- \bigcirc n to be a power of 2, then $\mathcal{A}(n)$ is the *required* collapsible set.

- lacksquare Can do all basic calculations on complex numbers in O(1) time.
- 2 Idea: Work over the complex numbers.
- Use roots of unity!
- \circ γ : nth root of unity. There are n such roots, and let $\gamma_j(n)$ denote the jth root.

$$\gamma_j(n) = \cos((2\pi j)/n) + \mathrm{i}\sin((2\pi j)/n) = \gamma^j.$$

Let
$$\mathcal{A}(n) = \{\gamma_0(n), \ldots, \gamma_{n-1}(n)\}.$$

- $| \mathsf{SQ}(\mathcal{A}(n)) |$ has n/2 entries.
- 0 n to be a power of 2, then $\mathcal{A}(n)$ is the *required* collapsible set.

- lacksquare Can do all basic calculations on complex numbers in O(1) time.
- ② Idea: Work over the complex numbers.
- Use roots of unity!
- \circ γ : nth root of unity. There are n such roots, and let $\gamma_j(n)$ denote the jth root.

$$\gamma_j(n) = \cos((2\pi j)/n) + \mathrm{i}\sin((2\pi j)/n) = \gamma^j.$$

Let
$$\mathcal{A}(n) = \{\gamma_0(n), \ldots, \gamma_{n-1}(n)\}.$$

- | SQ(A(n)) | has n/2 entries.
- ${\color{red}0}$ ${\color{red}n}$ to be a power of 2, then ${\color{red}\mathcal{A}(n)}$ is the *required* collapsible set.

- lacktriangle Can do all basic calculations on complex numbers in O(1) time.
- Idea: Work over the complex numbers.
- Use roots of unity!
- $oldsymbol{\circ}$ γ : nth root of unity. There are n such roots, and let $\gamma_j(n)$ denote the jth root.

$$\gamma_j(n) = \cos((2\pi j)/n) + \mathrm{i}\sin((2\pi j)/n) = \gamma^j.$$

Let
$$\mathcal{A}(n) = \{\gamma_0(n), \ldots, \gamma_{n-1}(n)\}.$$

- $| \mathsf{SQ}(\mathcal{A}(n)) |$ has n/2 entries.
- ${\color{red}0}$ ${\color{red}n}$ to be a power of 2, then ${\color{red}\mathcal{A}(n)}$ is the *required* collapsible set.

- lacksquare Can do all basic calculations on complex numbers in O(1) time.
- ② Idea: Work over the complex numbers.
- Use roots of unity!
- $oldsymbol{\circ}$ γ : nth root of unity. There are n such roots, and let $\gamma_j(n)$ denote the jth root.

$$\gamma_j(n) = \cos((2\pi j)/n) + \mathrm{i}\sin((2\pi j)/n) = \gamma^j.$$

Let
$$\mathcal{A}(n) = \{\gamma_0(n), \ldots, \gamma_{n-1}(n)\}.$$

- $|\mathbf{SQ}(\mathcal{A}(n))|$ has n/2 entries.
- $oldsymbol{0}$ n to be a power of 2, then $\mathcal{A}(n)$ is the *required* collapsible set.

- lacksquare Can do all basic calculations on complex numbers in O(1) time.
- ② Idea: Work over the complex numbers.
- Use roots of unity!
- $oldsymbol{\circ}$ γ : nth root of unity. There are n such roots, and let $\gamma_j(n)$ denote the jth root.

$$\gamma_j(n) = \cos((2\pi j)/n) + \mathrm{i}\sin((2\pi j)/n) = \gamma^j.$$

Let
$$\mathcal{A}(n) = \{\gamma_0(n), \ldots, \gamma_{n-1}(n)\}.$$

- $|\mathbf{SQ}(\mathcal{A}(n))|$ has n/2 entries.
- ullet SQ $(\mathcal{A}(n))=\mathcal{A}(n/2)$
- $oldsymbol{0}$ n to be a power of 2, then $\mathcal{A}(n)$ is the *required* collapsible set.

- lacktriangle Can do all basic calculations on complex numbers in O(1) time.
- ② Idea: Work over the complex numbers.
- Use roots of unity!
- $oldsymbol{\circ}$ γ : nth root of unity. There are n such roots, and let $\gamma_j(n)$ denote the jth root.

$$\gamma_j(n) = \cos((2\pi j)/n) + \mathrm{i}\sin((2\pi j)/n) = \gamma^j.$$

Let
$$\mathcal{A}(n) = \{\gamma_0(n), \ldots, \gamma_{n-1}(n)\}.$$

- |SQ($\mathcal{A}(n)$)| has n/2 entries.
- ullet SQ $(\mathcal{A}(n))=\mathcal{A}(n/2)$
- $m{0}$ n to be a power of $m{2}$, then $m{\mathcal{A}}(n)$ is the *required* collapsible set.

The first result...

Theorem

Given polynomial p(x) of degree n, where n is a power of two, then we can compute p(X) in $O(n \log n)$ time, where $X = \mathcal{A}(n)$ is the set of n different powers of the nth root of unity over the complex numbers.

- Can multiply two polynomials quickly
- 2 by transforming them to the point-value pairs representation...
- \bigcirc over the nth roots of unity.
- Q: How to transform this representation back to the regular representation.
- A: Do some confusing math...

- Can multiply two polynomials quickly
- by transforming them to the point-value pairs representation...
- \bigcirc over the nth roots of unity.
- Q: How to transform this representation back to the regular representation.
- A: Do some confusing math...

- Can multiply two polynomials quickly
- 2 by transforming them to the point-value pairs representation...
- $oldsymbol{0}$ over the nth roots of unity.
- Q: How to transform this representation back to the regular representation.
- A: Do some confusing math...

- Can multiply two polynomials quickly
- ② by transforming them to the point-value pairs representation...
- \odot over the nth roots of unity.
- Q: How to transform this representation back to the regular representation.
- A: Do some confusing math...

- Can multiply two polynomials quickly
- 2 by transforming them to the point-value pairs representation...
- over the nth roots of unity.
- Q: How to transform this representation back to the regular representation.
- A: Do some confusing math...

Recovering the polynomial

Think about FFT as a matrix multiplication operator. $p(x) = \sum_{i=0}^{n-1} a_i x^i$. Evaluating $p(\cdot)$ on $\mathcal{A}(n)$:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \gamma_0 & \gamma_0^2 & \gamma_0^3 & \cdots & \gamma_0^{n-1} \\ 1 & \gamma_1 & \gamma_1^2 & \gamma_1^3 & \cdots & \gamma_1^{n-1} \\ 1 & \gamma_2 & \gamma_2^2 & \gamma_2^3 & \cdots & \gamma_2^{n-1} \\ 1 & \gamma_3 & \gamma_3^2 & \gamma_3^3 & \cdots & \gamma_3^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \gamma_{n-1} & \gamma_{n-1}^2 & \gamma_{n-1}^3 & \cdots & \gamma_{n-1}^{n-1} \end{pmatrix}}_{\text{the matrix } V} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{n-1} \end{pmatrix},$$

where $\gamma_j = \gamma_j(n) = (\gamma_1(n))^j$ is the *j*th power of the *n*th root of unity, and $y_i = p(\gamma_i)$.

The Vandermonde matrix

Because every matrix needs a name

 $oldsymbol{V}$ is the $oldsymbol{Vandermonde}$ matrix.

 V^{-1} : inverse matrix of V

Vandermonde matrix. And let multiply the above formula from the left. We get:

$$egin{pmatrix} y_0 \ y_1 \ y_2 \ dots \ y_{n-1} \end{pmatrix} = V egin{pmatrix} a_0 \ a_1 \ a_2 \ a_3 \ dots \ a_{n-1} \end{pmatrix} \qquad \Longrightarrow \qquad egin{pmatrix} a_0 \ a_1 \ a_2 \ a_3 \ dots \ a_{n-1} \end{pmatrix} = V^{-1} egin{pmatrix} y_0 \ y_1 \ y_2 \ dots \ y_{n-1} \end{pmatrix}.$$

..for the rescue

$$\left\{(\gamma_0,p(\gamma_0)),(\gamma_1,p(\gamma_1)),\ldots,(\gamma_{n-1},p(\gamma_{n-1}))
ight\}$$

- ② by doing a single matrix multiplication of V^{-1} by the vector $[y_0, y_1, \dots, y_{n-1}]$.
- Multiplying a vector with n entries with $n \times n$ matrix takes $O(n^2)$ time.
- No benefit so far...

..for the rescue

$$\Big\{(\gamma_0,p(\gamma_0)),(\gamma_1,p(\gamma_1)),\ldots,(\gamma_{n-1},p(\gamma_{n-1}))\Big\}$$

- ② by doing a single matrix multiplication of V^{-1} by the vector $[y_0,y_1,\ldots,y_{n-1}].$
- Multiplying a vector with n entries with $n \times n$ matrix takes $O(n^2)$ time.
- No benefit so far...

..for the rescue

$$\Big\{(\gamma_0,p(\gamma_0)),(\gamma_1,p(\gamma_1)),\ldots,(\gamma_{n-1},p(\gamma_{n-1}))\Big\}$$

- ② by doing a single matrix multiplication of V^{-1} by the vector $[y_0,y_1,\ldots,y_{n-1}]$.
- Multiplying a vector with n entries with $n \times n$ matrix takes $O(n^2)$ time.
- No benefit so far...

..for the rescue

$$\Big\{(\gamma_0,p(\gamma_0)),(\gamma_1,p(\gamma_1)),\ldots,(\gamma_{n-1},p(\gamma_{n-1}))\Big\}$$

- ② by doing a single matrix multiplication of V^{-1} by the vector $[y_0,y_1,\ldots,y_{n-1}]$.
- Multiplying a vector with n entries with $n \times n$ matrix takes $O(n^2)$ time.
- No benefit so far...

What is the inverse of the Vandermonde matrix

Vandermonde matrix is famous, beautiful and well known - a celebrity matrix

Claim

$$V^{-1} = \frac{1}{n} \begin{pmatrix} 1 & \beta_0 & \beta_0^2 & \beta_0^3 & \cdots & \beta_0^{n-1} \\ 1 & \beta_1 & \beta_1^2 & \beta_1^3 & \cdots & \beta_1^{n-1} \\ 1 & \beta_2 & \beta_2^2 & \beta_2^3 & \cdots & \beta_2^{n-1} \\ 1 & \beta_3 & \beta_3^2 & \beta_3^3 & \cdots & \beta_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \beta_{n-1} & \beta_{n-1}^2 & \beta_{n-1}^3 & \cdots & \beta_{n-1}^{n-1} \end{pmatrix},$$

where $\beta_j = (\gamma_j(n))^{-1}$.

Proof

Consider the (u,v) entry in the matrix $C=V^{-1}\,V$. We have

$$C_{u,v} = \sum_{j=0}^{n-1} rac{(eta_u)^j (\gamma_j)^v}{n}.$$

As $\gamma_j=(\gamma_1)^j$. Thus,

$$C_{u,v} = \sum_{j=0}^{n-1} rac{(eta_u)^j ((\gamma_1)^j)^v}{n} = \sum_{j=0}^{n-1} rac{(eta_u)^j ((\gamma_1)^v)^j}{n} = \sum_{j=0}^{n-1} rac{(eta_u\gamma_v)^j}{n}.$$

Clearly, if $oldsymbol{u} = oldsymbol{v}$ then

$$C_{u,u} = rac{1}{n} \sum_{i=0}^{n-1} (eta_u \gamma_u)^j = rac{1}{n} \sum_{i=0}^{n-1} (1)^j = rac{n}{n} = 1.$$

Proof continued...

If $u \neq v$ then,

$$eta_u\gamma_v=(\gamma_u)^{-1}\gamma_v=(\gamma_1)^{-u}\gamma_1^v=(\gamma_1)^{v-u}=\gamma_{v-u}.$$

And

$$C_{u,v} = rac{1}{n} \sum_{j=0}^{n-1} (\gamma_{v-u})^j = rac{1}{n} \cdot rac{\gamma_{v-u}^n - 1}{\gamma_{v-u} - 1} = rac{1}{n} \cdot rac{1-1}{\gamma_{v-u} - 1} = 0,$$

Proved that the matrix $oldsymbol{C}$ have ones on the diagonal and zero everywhere else.

Recap...

- ① n point-value pairs $\{(\gamma_0,y_0),\ldots,(\gamma_{n-1},y_{n-1})\}$: of polynomial $p(x)=\sum_{i=0}^{n-1}a_ix^i$ over nth roots of unity.
- 2 Recover coefficients of polynomial by multiplying $[y_0, y_1, \ldots, y_n]$ by V^{-1} :

$$egin{pmatrix} a_0 \ a_1 \ a_2 \ dots \ a_{n-1} \end{pmatrix} = \underbrace{rac{1}{n}egin{pmatrix} eta_0 & eta_0^2 & eta_0^3 & \cdots & eta_0^{n-1} \ 1 & eta_1 & eta_1^2 & eta_1^3 & \cdots & eta_1^{n-1} \ 1 & eta_2 & eta_2^2 & eta_2^3 & \cdots & eta_2^{n-1} \ 1 & eta_3 & eta_3^2 & eta_3^3 & \cdots & eta_2^{n-1} \ 1 & eta_3 & eta_3^2 & eta_3^3 & \cdots & eta_3^{n-1} \ dots & dots & dots & dots & dots \ 1 & eta_{n-1} & eta_{n-1}^2 & eta_{n-1}^3 & \cdots & eta_{n-1}^{n-1} \end{pmatrix}} egin{pmatrix} y_0 \ y_1 \ y_2 \ y_3 \ dots \ y_{n-1} \$$

$$oldsymbol{W}(x) = \sum\limits_{i=0}^{n-1} (y_i/n) x^i$$
: $a_i = W(eta_i)$.

Sariel (UIUC) CS573 31 Fall 2014 31 / 40

Recap...

- ① n point-value pairs $\{(\gamma_0,y_0),\ldots,(\gamma_{n-1},y_{n-1})\}$: of polynomial $p(x)=\sum_{i=0}^{n-1}a_ix^i$ over nth roots of unity.
- **2** Recover coefficients of polynomial by multiplying $[y_0, y_1, \ldots, y_n]$ by V^{-1} :

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{pmatrix} = \underbrace{\frac{1}{n} \begin{pmatrix} 1 & \beta_0 & \beta_0^2 & \beta_0^3 & \cdots & \beta_0^{n-1} \\ 1 & \beta_1 & \beta_1^2 & \beta_1^3 & \cdots & \beta_1^{n-1} \\ 1 & \beta_2 & \beta_2^2 & \beta_2^3 & \cdots & \beta_2^{n-1} \\ 1 & \beta_3 & \beta_3^2 & \beta_3^3 & \cdots & \beta_3^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \beta_{n-1} & \beta_{n-1}^2 & \beta_{n-1}^3 & \cdots & \beta_{n-1}^{n-1} \end{pmatrix}}_{V^{-1}} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \end{pmatrix}.$$

$$ext{ } ext{ } ext$$

Sariel (UIUC) CS573 31 Fall 2014 31 / 40

Recap...

- ① n point-value pairs $\{(\gamma_0, y_0), \ldots, (\gamma_{n-1}, y_{n-1})\}$: of polynomial $p(x) = \sum_{i=0}^{n-1} a_i x^i$ over nth roots of unity.
- **2** Recover coefficients of polynomial by multiplying $[y_0, y_1, \ldots, y_n]$ by V^{-1} :

$$egin{pmatrix} a_0 \ a_1 \ a_2 \ dots \ a_{n-1} \end{pmatrix} = \underbrace{ egin{pmatrix} 1 & eta_0 & eta_0^2 & eta_0^3 & \cdots & eta_0^{n-1} \ 1 & eta_1 & eta_1^2 & eta_1^3 & \cdots & eta_1^{n-1} \ 1 & eta_2 & eta_2^2 & eta_2^3 & \cdots & eta_2^{n-1} \ 1 & eta_3 & eta_3^2 & eta_3^3 & \cdots & eta_3^{n-1} \ dots & dots & dots & dots & dots & dots \ 1 & eta_{n-1} & eta_{n-1}^2 & eta_{n-1}^3 & \cdots & eta_{n-1}^{n-1} \end{pmatrix} } egin{pmatrix} y_0 \ y_1 \ y_2 \ y_3 \ dots \ y_{n-1} \end{pmatrix}.$$

$$oldsymbol{W}(x) = \sum\limits_{i=0}^{n-1} (y_i/n) x^i$$
: $a_i = W(eta_i)$.

Sariel (UIUC) CS573 31 Fall 2014 31 / 40

- recover coefficients of $p(\cdot)$...
- ② ... compute $W(\cdot)$ on n values: $\beta_0, \ldots, \beta_{n-1}$.
- ① Indeed $\beta_i^n = (\gamma_i^{-1})^n = (\gamma_i^n)^{-1} = 1^{-1} = 1$.
- ullet Apply the **FFTAlg** algorithm on W(x) to compute a_0,\ldots,a_{n-1} .

- recover coefficients of $p(\cdot)$...
- $oldsymbol{0}$... compute $W(\cdot)$ on n values: eta_0,\ldots,eta_{n-1} .
- ① Indeed $\beta_i^n = (\gamma_i^{-1})^n = (\gamma_i^n)^{-1} = 1^{-1} = 1$.
- ullet Apply the **FFTAlg** algorithm on W(x) to compute a_0,\ldots,a_{n-1} .

- recover coefficients of $p(\cdot)$...
- \bullet ... compute $W(\cdot)$ on n values: $\beta_0, \ldots, \beta_{n-1}$.
- ① Indeed $\beta_i^n = (\gamma_i^{-1})^n = (\gamma_i^n)^{-1} = 1^{-1} = 1$.
- ullet Apply the **FFTAlg** algorithm on W(x) to compute a_0,\ldots,a_{n-1} .

- recover coefficients of $p(\cdot)$...
- \bullet ... compute $W(\cdot)$ on n values: $\beta_0, \ldots, \beta_{n-1}$.

- ullet Apply the **FFTAlg** algorithm on W(x) to compute a_0,\ldots,a_{n-1} .

- recover coefficients of $p(\cdot)$...

- ullet Apply the **FFTAlg** algorithm on W(x) to compute a_0,\ldots,a_{n-1} .

Result

Theorem

Given n point-value pairs of a polynomial p(x) of degree n-1 over the set of n powers of the nth roots of unity, we can recover the polynomial p(x) in $O(n \log n)$ time.

Theorem

Given two polynomials of degree n, they can be multiplied in $O(n \log n)$ time.

Sariel (UIUC) CS573 33 Fall 2014 33 / 40

Part II

- lacksquare Two vectors: $A=[a_0,a_1,\ldots,a_n]$ and $B=[b_0,\ldots,b_n]$.
- ② dot product $A \cdot B = \langle A, B \rangle = \sum_{i=0}^n a_i b_i$.
- ① Padded with zeros:, $a_j=0$ for $j \notin \{0,\ldots,n\}$).
- $egin{aligned} oldsymbol{a}_r &= ig[a_{n-r}, a_{n+1-r}, a_{n+2-r}, \ldots, a_{2n-r} ig] \ \end{aligned}$ where $a_j = 0$ if $j
 otin ig[0, \ldots, n ig]$.
- **Observation**: $A_n = A$.

- lacksquare Two vectors: $A=[a_0,a_1,\ldots,a_n]$ and $B=[b_0,\ldots,b_n]$.
- ② dot product $A \cdot B = \langle A, B \rangle = \sum_{i=0}^n a_i b_i$.
- $lacksquare A_r$: shifting of A by n-r locations to the left
- ① Padded with zeros:, $a_j = 0$ for $j \notin \{0, \ldots, n\}$).
- $egin{aligned} oldsymbol{a}_r &= ig[a_{n-r}, a_{n+1-r}, a_{n+2-r}, \ldots, a_{2n-r} ig] \ \end{aligned}$ where $a_j = 0$ if $j
 otin ig[0, \ldots, n ig]$.
- **Observation**: $A_n = A$.

- lacksquare Two vectors: $A=[a_0,a_1,\ldots,a_n]$ and $B=[b_0,\ldots,b_n]$.
- **2** dot product $A \cdot B = \langle A, B \rangle = \sum_{i=0}^n a_i b_i$.
- $lacksquare A_r$: shifting of A by n-r locations to the left
- ① Padded with zeros:, $a_j = 0$ for $j \notin \{0, \ldots, n\}$).
- $egin{aligned} oldsymbol{a}_r &= ig[a_{n-r}, a_{n+1-r}, a_{n+2-r}, \ldots, a_{2n-r} ig] \ \end{aligned}$ where $a_j = 0$ if $oldsymbol{j}
 otin ig[0, \ldots, n ig]$.
- **Observation**: $A_n = A$.

- lacksquare Two vectors: $A=[a_0,a_1,\ldots,a_n]$ and $B=[b_0,\ldots,b_n]$.
- $m{@}$ dot product $A\cdot B=\langle A,B
 angle =\sum_{i=0}^n a_ib_i.$
- $lacksquare A_r$: shifting of A by n-r locations to the left
- **1** Padded with zeros:, $a_j = 0$ for $j \notin \{0, \ldots, n\}$).
- $egin{aligned} oldsymbol{a}_r &= ig[a_{n-r}, a_{n+1-r}, a_{n+2-r}, \ldots, a_{2n-r} ig] \ \end{aligned}$ where $a_j = 0$ if $j
 otin ig[0, \ldots, n ig]$.
- **Observation**: $A_n = A$.

Convolutions

- lacksquare Two vectors: $A=[a_0,a_1,\ldots,a_n]$ and $B=[b_0,\ldots,b_n]$.
- **2** dot product $A \cdot B = \langle A, B \rangle = \sum_{i=0}^n a_i b_i$.
- $lacksquare A_r$: shifting of A by n-r locations to the left
- **1** Padded with zeros:, $a_j = 0$ for $j \notin \{0, \ldots, n\}$).
- $oldsymbol{a}_r = egin{bmatrix} a_{n-r}, a_{n+1-r}, a_{n+2-r}, \ldots, a_{2n-r} \end{bmatrix}$ where $a_j = 0$ if $j
 otin oldsymbol{0}, \ldots, n igcap$.
- **Observation**: $A_n = A$.

Convolutions

- lacksquare Two vectors: $A=[a_0,a_1,\ldots,a_n]$ and $B=[b_0,\ldots,b_n]$.
- **②** dot product $A \cdot B = \langle A, B \rangle = \sum_{i=0}^n a_i b_i$.
- $lacksquare A_r$: shifting of A by n-r locations to the left
- lacksquare Padded with zeros:, $a_j=0$ for $j
 otin\{0,\ldots,n\}$).
- $oldsymbol{a}_r = ig[a_{n-r}, a_{n+1-r}, a_{n+2-r}, \ldots, a_{2n-r}ig]$ where $a_j = 0$ if $j
 otin ig[0, \ldots, nig]$.
- **10 Observation**: $A_n = A$.

Example of shifting

Example

For
$$A=[3,7,9,15]$$
, $n=3$ $A_2=[7,9,15,0]$, $A_5=[0,0,3,7]$.

Definition

Definition

Let $c_i = A_i \cdot B = \sum_{j=n-i}^{2n-i} a_j b_{j-n+i}$, for $i = 0, \ldots, 2n$. The vector $[c_0, \ldots, c_{2n}]$ is the **convolution** of A and B.

question

How to compute the convolution of two vectors of length n?

Sariel (UIUC) CS573 37 Fall 2014 37 / 40

- $lackbox{0} \ p(x) = \sum_{i=0}^n lpha_i x^i$, and $q(x) = \sum_{i=0}^n eta_i x^i$.
- ② Coefficient of x^i in r(x) = p(x)q(x) is $d_i = \sum_{j=0}^i \alpha_j \beta_{i-j}$.
- ① Want to compute $c_i = A_i \cdot B = \sum_{j=n-i}^{2n-i} a_j b_{j-n+i}$.
- ullet Set $lpha_i=a_i$ and $eta_l=b_{n-l-1}$.

- $lackbox{0} \ p(x) = \sum_{i=0}^n lpha_i x^i$, and $q(x) = \sum_{i=0}^n eta_i x^i$.
- ② Coefficient of x^i in r(x) = p(x)q(x) is $d_i = \sum_{j=0}^i \alpha_j \beta_{i-j}$.
- ① Want to compute $c_i = A_i \cdot B = \sum_{j=n-i}^{2n-i} a_j b_{j-n+i}$.
- $lack {f O}$ Set $lpha_i=a_i$ and $eta_l=b_{n-l-1}$

- $lacksquare p(x) = \sum_{i=0}^n lpha_i x^i$, and $q(x) = \sum_{i=0}^n eta_i x^i$.
- ② Coefficient of x^i in r(x) = p(x)q(x) is $d_i = \sum_{j=0}^i \alpha_j \beta_{i-j}$.
- ullet Want to compute $c_i = A_i \cdot B = \sum_{j=n-i}^{2n-i} a_j b_{j-n+i}$.
- ullet Set $lpha_i=a_i$ and $eta_l=b_{n-l-1}$

- $oldsymbol{0} p(x) = \sum_{i=0}^n lpha_i x^i$, and $q(x) = \sum_{i=0}^n eta_i x^i$.
- ② Coefficient of x^i in r(x) = p(x)q(x) is $d_i = \sum_{j=0}^i \alpha_j \beta_{i-j}$.
- **3** Want to compute $c_i = A_i \cdot B = \sum_{j=n-i}^{2n-i} a_j b_{j-n+i}$.
- lacktriangledown Set $lpha_i=a_i$ and $eta_l=b_{n-l-1}$.

Convolution by example

- Consider coefficient of x^2 in product of $p(x)=a_0+a_1x+a_2x^2+a_3x^3$ and $q(x)=b_0+b_1x+b_2x^2+b_3x^3$.
- Sum of the entries on the anti diagonal

	a_0+	a_1x	$+a_2x^2$	$+a_{3}x^{3}$
b_0			$a_2b_0x^2$	
$+b_1x$		$a_1b_1x^2$		
$+b_{2}x^{2}$	$a_0b_2x^2$			
$+b_3x^3$				

ullet entry in the ith row and jth column is a_ib_j .

Convolution by example

- ① Consider coefficient of x^2 in product of $p(x)=a_0+a_1x+a_2x^2+a_3x^3$ and $q(x)=b_0+b_1x+b_2x^2+b_3x^3$.
- Sum of the entries on the anti diagonal:

	a_0+	a_1x	$ +a_2x^2 $	$+a_3x^3$
b_0			$a_2b_0x^2$	
$+b_1x$		$a_1b_1x^2$		
$+b_2x^2$	$a_0b_2x^2$			
$+b_3x^3$				

 \bigcirc entry in the *i*th row and *j*th column is $a_i b_j$.

Convolution by example

- ① Consider coefficient of x^2 in product of $p(x)=a_0+a_1x+a_2x^2+a_3x^3$ and $q(x)=b_0+b_1x+b_2x^2+b_3x^3$.
- Sum of the entries on the anti diagonal:

	a_0+	a_1x	$+a_2x^2$	$+a_3x^3$
b_0			$a_2b_0x^2$	
$+b_1x$		$a_1b_1x^2$		
$+b_{2}x^{2}$	$a_0b_2x^2$			
$+b_{3}x^{3}$				

3 entry in the *i*th row and *j*th column is $a_i b_j$.

Convolution

Theorem

Given two vectors $A = [a_0, a_1, \ldots, a_n]$, $B = [b_0, \ldots, b_n]$ one can compute their convolution in $O(n \log n)$ time.

Proof.

Let $p(x) = \sum_{i=0}^n a_{n-i} x^i$ and let $q(x) = \sum_{i=0}^n b_i x^i$. Compute r(x) = p(x) q(x) in $O(n \log n)$ time using the convolution theorem. Let c_0, \ldots, c_{2n} be the coefficients of r(x). It is easy to verify, as described above, that $[c_0, \ldots, c_{2n}]$ is the convolution of A and B.

Sariel (UIUC) CS573 41 Fall 2014 41 / 40

Sariel (UIUC) CS573 42 Fall 2014 42 / 40

Sariel (UIUC) CS573 43 Fall 2014 43 / 40

Sariel (UIUC) CS573 44 Fall 2014 44 / 40