CSCI 5451 Fall 2015

Week 11 Notes

Professor Ellen Gethner

November 2, 2015

Fast Fourier Transform (FFT)

- The FFT has many practical applications, while being aesthetic at the same time.

Fast Fourier Transform (FFT)

- The FFT has many practical applications, while being aesthetic at the same time.
- The ideas and implications that stem from the FFT accelerated the foundations of computer science in the 1960s when the algorithm was first discovered.

Fast Fourier Transform (FFT)

- The FFT has many practical applications, while being aesthetic at the same time.
- The ideas and implications that stem from the FFT accelerated the foundations of computer science in the 1960s when the algorithm was first discovered.
- We'll study the FFT with one application (two, if time permits).

Fast Fourier Transform (FFT)

- The FFT has many practical applications, while being aesthetic at the same time.
- The ideas and implications that stem from the FFT accelerated the foundations of computer science in the 1960s when the algorithm was first discovered.
- We'll study the FFT with one application (two, if time permits).
- **FFT Application:** Polynomial multiplication (keyword= convolution for other applications).

FFT and Polynomial Multiplicaiton

- **Problem.** Given two polynomials $p(x)$ and $q(x)$, find the product $p(x)q(x)$.

FFT and Polynomial Multiplicaiton

- **Problem.** Given two polynomials $p(x)$ and $q(x)$, find the product $p(x)q(x)$.
- Naively, if both p and q have degree $n-1$, we can compute $p(x) \times q(x)$ in $O(n^2)$ time.

FFT and Polynomial Multiplicaiton

- **Problem.** Given two polynomials $p(x)$ and $q(x)$, find the product $p(x)q(x)$.
- Naively, if both p and q have degree $n-1$, we can compute $p(x) \times q(x)$ in $O(n^2)$ time.
- That is, if $p(x) = a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ and $q(x) = b_{n-1}x^{n-1} + \cdots + b_1 x + b_0$ then the straightforward multiplication would require $O(n^2)$ multiplications.

FFT and Polynomial Multiplicaiton

- **Problem.** Given two polynomials $p(x)$ and $q(x)$, find the product $p(x)q(x)$.
- Naively, if both p and q have degree $n-1$, we can compute $p(x) \times q(x)$ in $O(n^2)$ time.
- That is, if $p(x) = a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ and $q(x) = b_{n-1}x^{n-1} + \cdots + b_1 x + b_0$ then the straightforward multiplication would require $O(n^2)$ multiplications.
- The above method is more than adequate, but there are many applications that require real-time dynamic computations (such as interactive rendering of 3D graphics, for example)

FFT and Polynomial Multiplicaiton

- **Problem.** Given two polynomials $p(x)$ and $q(x)$, find the product $p(x)q(x)$.
- Naively, if both p and q have degree $n-1$, we can compute $p(x) \times q(x)$ in $O(n^2)$ time.
- That is, if $p(x) = a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ and $q(x) = b_{n-1}x^{n-1} + \cdots + b_1 x + b_0$ then the straightforward multiplication would require $O(n^2)$ multiplications.
- The above method is more than adequate, but there are many applications that require real-time dynamic computations (such as interactive rendering of 3D graphics, for example)
- and thus any reduction we can make in the complexity will be quite useful.

A Different Approach to Polynomial Multiplication

- **Representing a Polynomial.** The typical representation of $p(x)$ would be as a $1 \times n$ array $[a_{n-1}, a_{n-2}, \ldots, a_1, a_0]$.

A Different Approach to Polynomial Multiplication

- **Representing a Polynomial.** The typical representation of $p(x)$ would be as a $1 \times n$ array $[a_{n-1}, a_{n-2}, \ldots, a_1, a_0]$.

- **Thinking outside of the box.** Consider a polynomial of degree 1 (ie, a linear polynomial): $\ell(x) = a_1 x + a_0$.

A Different Approach to Polynomial Multiplication

- **Representing a Polynomial.** The typical representation of $p(x)$ would be as a $1 \times n$ array $[a_{n-1}, a_{n-2}, \ldots, a_1, a_0]$.

- **Thinking outside of the box.** Consider a polynomial of degree 1 (ie, a linear polynomial): $\ell(x) = a_1 x + a_0$.

- We can represent $\ell(x)$ by way of its coefficients (been there, done that) or

A Different Approach to Polynomial Multiplication

- **Representing a Polynomial.** The typical representation of $p(x)$ would be as a $1 \times n$ array $[a_{n-1}, a_{n-2}, \ldots, a_1, a_0]$.
- **Thinking outside of the box.** Consider a polynomial of degree 1 (ie, a linear polynomial): $\ell(x) = a_1 x + a_0$.
- We can represent $\ell(x)$ by way of its coefficients (been there, done that) or
- instead we could represent $\ell(x)$ by any two distinct points on $\ell(x)$.

A Different Approach to Polynomial Multiplication

- **Representing a Polynomial.** The typical representation of $p(x)$ would be as a $1 \times n$ array $[a_{n-1}, a_{n-2}, \ldots, a_1, a_0]$.
- **Thinking outside of the box.** Consider a polynomial of degree 1 (ie, a linear polynomial): $\ell(x) = a_1 x + a_0$.
- We can represent $\ell(x)$ by way of its coefficients (been there, done that) or
- instead we could represent $\ell(x)$ by any two distinct points on $\ell(x)$.
- Another way of making the above point (no pun intended) is to remember that **two points uniquely determine a line**."

Two points determine a line and...

- More generally, a polynomial $p(x)$ of degree $n-1$ is uniquely determined by any n distinct points on the curve $p(x)$.

Two points determine a line and...

- More generally, a polynomial $p(x)$ of degree $n-1$ is uniquely determined by any n distinct points on the curve $p(x)$.
- Why? Because you can solve for the n coefficients of p with n equations and n unknowns as usual by algebra.

Two points determine a line and...

- More generally, a polynomial $p(x)$ of degree $n-1$ is uniquely determined by any n distinct points on the curve $p(x)$.

- Why? Because you can solve for the n coefficients of p with n equations and n unknowns as usual by algebra.

- **Example.** Suppose $p(x) = a_2x^2 + a_1x + a_0$ and that points (3,65), (1, 41), and (2, 57) are all on the curve determined by $p(x)$.

Two points determine a line and...

- More generally, a polynomial $p(x)$ of degree $n-1$ is uniquely determined by any n distinct points on the curve $p(x)$.

- Why? Because you can solve for the n coefficients of p with n equations and n unknowns as usual by algebra.

- **Example.** Suppose $p(x) = a_2x^2 + a_1x + a_0$ and that points (3,65), (1, 41), and (2, 57) are all on the curve determined by $p(x)$.

- Find the coefficients of $p(x)$. That is, solve for a_2, a_1, and a_0.

Two points determine a line and...

- More generally, a polynomial $p(x)$ of degree $n-1$ is uniquely determined by any n distinct points on the curve $p(x)$.
- Why? Because you can solve for the n coefficients of p with n equations and n unknowns as usual by algebra.
- **Example.** Suppose $p(x) = a_2x^2 + a_1x + a_0$ and that points (3,65), (1, 41), and (2, 57) are all on the curve determined by $p(x)$.
- Find the coefficients of $p(x)$. That is, solve for a_2, a_1, and a_0.
- We have $67 = a_23^2 + a_13 + a_0$, $51 = a_21^2 + a_11 + a_0$, and $57 = a_22^2 + a_12 + a_0$, and thus we have three equations in three unknowns.

Example, continued

- I used *Mathematica* and the command **LinearSolve** to solve for the coefficients of $p(x)$:

Example, continued

- I used *Mathematica* and the command **LinearSolve** to solve for the coefficients of $p(x)$:

In[25]:=
```
Clear[matrixA]; matrixA = {{9, 3, 1}, {1, 1, 1}, {4, 2, 1}};
matrixA // MatrixForm
```

Out[26]//MatrixForm=

$$\begin{pmatrix} 9 & 3 & 1 \\ 1 & 1 & 1 \\ 4 & 2 & 1 \end{pmatrix}$$

In[27]:=
```
Clear[vectorOfAnswers]; vectorOfAnswers = {65, 51, 57};
vectorOfAnswers // MatrixForm
```

Out[28]//MatrixForm=

$$\begin{pmatrix} 65 \\ 51 \\ 57 \end{pmatrix}$$

In[29]:=
```
LinearSolve[matrixA, vectorOfAnswers]
```

Out[29]= {1, 3, 47}

Example, continued

- I used *Mathematica* and the command **LinearSolve** to solve for the coefficients of $p(x)$:

```
In[25]:= Clear[matrixA]; matrixA = {{9, 3, 1}, {1, 1, 1}, {4, 2, 1}};
         matrixA // MatrixForm
```

Out[26]//MatrixForm=

$$\begin{pmatrix} 9 & 3 & 1 \\ 1 & 1 & 1 \\ 4 & 2 & 1 \end{pmatrix}$$

```
In[27]:=
         Clear[vectorOfAnswers]; vectorOfAnswers = {65, 51, 57};
         vectorOfAnswers // MatrixForm
```

Out[28]//MatrixForm=

$$\begin{pmatrix} 65 \\ 51 \\ 57 \end{pmatrix}$$

```
In[29]:= LinearSolve[matrixA, vectorOfAnswers]

Out[29]= {1, 3, 47}
```

- Thus $p(x) = x^2 + 3x + 47$.

Another Example and the FFT Journey Continued

- **Example.** The polynomial $p(x) = x^2 + 3x + 1$ is uniquely determined by the points (1,5), (2, 11), and (3,19) as well as many other choices of three points.

Another Example and the FFT Journey Continued

- **Example.** The polynomial $p(x) = x^2 + 3x + 1$ is uniquely determined by the points (1,5), (2, 11), and (3,19) as well as many other choices of three points.

- **Quick Check:**

Another Example and the FFT Journey Continued

- **Example.** The polynomial $p(x) = x^2 + 3x + 1$ is uniquely determined by the points (1,5), (2, 11), and (3,19) as well as many other choices of three points.

- **Quick Check:**

```
In[34]:= p[x_] := x^2 + 3 x + 1

In[35]:= p[1] == 5

Out[35]= True

In[36]:= p[2] == 11

Out[36]= True

In[37]:= p[3] == 19

Out[37]= True
```

A new idea!

A new idea!

- Maybe we can change the idea of multiplying polynomials to that of “multiplying points” and save some time.

A new idea!

- Maybe we can change the idea of multiplying polynomials to that of "multiplying points" and save some time.
- For example, $q(x) = 2x^2 - x + 3$ is uniquely determined by points (1,4), (2,9), and (3,18), which means

A new idea!

- Maybe we can change the idea of multiplying polynomials to that of "multiplying points" and save some time.
- For example, $q(x) = 2x^2 - x + 3$ is uniquely determined by points (1,4), (2,9), and (3,18), which means
- the product $p(x)q(x)$ contains points (1,20), (2, 99), and (3, 342).

A new idea!

- Maybe we can change the idea of multiplying polynomials to that of "multiplying points" and save some time.
- For example, $q(x) = 2x^2 - x + 3$ is uniquely determined by points (1,4), (2,9), and (3,18), which means
- the product $p(x)q(x)$ contains points (1,20), (2, 99), and (3, 342).
- Why?

A new idea!

- Maybe we can change the idea of multiplying polynomials to that of "multiplying points" and save some time.
- For example, $q(x) = 2x^2 - x + 3$ is uniquely determined by points (1,4), (2,9), and (3,18), which means
- the product $p(x)q(x)$ contains points (1,20), (2, 99), and (3, 342).
- Why?
- But $p(x)q(x) = 2x^4 + 5x^3 + 2x^2 + 8x + 3$ and so is not uniquely determined by the three points above.

A new idea!

- Maybe we can change the idea of multiplying polynomials to that of "multiplying points" and save some time.
- For example, $q(x) = 2x^2 - x + 3$ is uniquely determined by points (1,4), (2,9), and (3,18), which means
- the product $p(x)q(x)$ contains points (1,20), (2, 99), and (3, 342).
- Why?
- But $p(x)q(x) = 2x^4 + 5x^3 + 2x^2 + 8x + 3$ and so is not uniquely determined by the three points above.
- In particular, we must represent the above polynomial of degree four by five points.

The solution

- Notice that the points that we choose to represent each of $p(x)$ and $q(x)$ must have matching x-coordinates. Why?

The solution

- Notice that the points that we choose to represent each of $p(x)$ and $q(x)$ must have matching x-coordinates. Why?
- Represent $p(x) = x^2 + 3x + 1$ and $q(x) = 2x^2 - x + 3$ by five points each.

The solution

- Notice that the points that we choose to represent each of $p(x)$ and $q(x)$ must have matching x-coordinates. Why?
- Represent $p(x) = x^2 + 3x + 1$ and $q(x) = 2x^2 - x + 3$ by five points each.
- We'll add (0,1) and (-1,-1) to the set of points representing $p(x)$.

The solution

- Notice that the points that we choose to represent each of $p(x)$ and $q(x)$ must have matching x-coordinates. Why?
- Represent $p(x) = x^2 + 3x + 1$ and $q(x) = 2x^2 - x + 3$ by five points each.
- We'll add (0,1) and (-1,-1) to the set of points representing $p(x)$.
- And we'll add (0,3) and (-1,6) to the set of points representing $q(x)$.

The solution

- Notice that the points that we choose to represent each of $p(x)$ and $q(x)$ must have matching x-coordinates. Why?
- Represent $p(x) = x^2 + 3x + 1$ and $q(x) = 2x^2 - x + 3$ by five points each.
- We'll add (0,1) and (-1,-1) to the set of points representing $p(x)$.
- And we'll add (0,3) and (-1,6) to the set of points representing $q(x)$.
- Thus the five points that we'll use to represent $p(x)q(x)$ are (1,20), (2,99), (3,342), (0,3), and (-1,-6).

The solution, continued

- The whole process of getting the 5-point representation of $p(x)q(x)$ only takes five scalar multiplications,

The solution, continued

- The whole process of getting the 5-point representation of $p(x)q(x)$ only takes five scalar multiplications,
- which is way better than the brute force method of 25 multiplications, additions, etc.

The solution, continued

- The whole process of getting the 5-point representation of $p(x)q(x)$ only takes five scalar multiplications,
- which is way better than the brute force method of 25 multiplications, additions, etc.
- **Our insight thus far:** We need an efficient method both for converting from points on a curve representing a polynomial

The solution, continued

- The whole process of getting the 5-point representation of $p(x)q(x)$ only takes five scalar multiplications,
- which is way better than the brute force method of 25 multiplications, additions, etc.
- **Our insight thus far:** We need an efficient method both for converting from points on a curve representing a polynomial
- **and** of evaluating polynomials at those points.

The solution, continued

- The whole process of getting the 5-point representation of $p(x)q(x)$ only takes five scalar multiplications,
- which is way better than the brute force method of 25 multiplications, additions, etc.
- **Our insight thus far:** We need an efficient method both for converting from points on a curve representing a polynomial
- **and** of evaluating polynomials at those points.
- The ideas above are the foundations of the FFT: it accomplishes both tasks efficiently.

Restatement of the Problem

- **Problem (again).** How can we evaluate two polynomials $p(x)$ and $q(x)$ of degree $n-1$, each at $2n-1$ distinct x-values so that the coefficients of the product polynomial $p(x)q(x)$ can be determined?

Restatement of the Problem

- **Problem (again).** How can we evaluate two polynomials $p(x)$ and $q(x)$ of degree $n-1$, each at $2n-1$ distinct x-values so that the coefficients of the product polynomial $p(x)q(x)$ can be determined?

- **Question.** Why WLOG can we assume that both polynomials have the same degree?

Restatement of the Problem

- **Problem (again).** How can we evaluate two polynomials $p(x)$ and $q(x)$ of degree $n-1$, each at $2n-1$ distinct x-values so that the coefficients of the product polynomial $p(x)q(x)$ can be determined?

- **Question.** Why WLOG can we assume that both polynomials have the same degree?

- **Answer.** If not, pad the coefficients of the lower degree polynomial with zeros.

Restatement of the Problem

- **Problem (again).** How can we evaluate two polynomials $p(x)$ and $q(x)$ of degree $n-1$, each at $2n-1$ distinct x-values so that the coefficients of the product polynomial $p(x)q(x)$ can be determined?

- **Question.** Why WLOG can we assume that both polynomials have the same degree?

- **Answer.** If not, pad the coefficients of the lower degree polynomial with zeros.

- In fact, by the same reasoning, our polynomials of degree $n-1$ can be viewed as polynomials of degree $2n-2$.

Restatement of the Problem

- **Problem (again).** How can we evaluate two polynomials $p(x)$ and $q(x)$ of degree $n-1$, each at $2n-1$ distinct x-values so that the coefficients of the product polynomial $p(x)q(x)$ can be determined?
- **Question.** Why WLOG can we assume that both polynomials have the same degree?
- **Answer.** If not, pad the coefficients of the lower degree polynomial with zeros.
- In fact, by the same reasoning, our polynomials of degree $n-1$ can be viewed as polynomials of degree $2n-2$.
- **One more restatement.** Evaluate an arbitrary polynomial $p(x) = \sum_{i=1}^{n-1} a_i x^i$ of degree $n-1$ at n distinct points.

Another Simplification

- **Set-up and notation.** For simplicity in upcoming arguments, we'll assume WLOG that n is a power of 2. (Why is this WLOG?)

Another Simplification

- **Set-up and notation.** For simplicity in upcoming arguments, we'll assume WLOG that n is a power of 2. (Why is this WLOG?)

- Matrix notation will greatly simplify our approach to the problem at hand.

Another Simplification

- **Set-up and notation.** For simplicity in upcoming arguments, we'll assume WLOG that n is a power of 2. (Why is this WLOG?)

- Matrix notation will greatly simplify our approach to the problem at hand.

- **Magic.** We magically choose $x_0, x_1, \ldots, x_{n-1}$ as the special distinct x-values at which to evaluate $p(x)$.

Another Simplification

- **Set-up and notation.** For simplicity in upcoming arguments, we'll assume WLOG that n is a power of 2. (Why is this WLOG?)
- Matrix notation will greatly simplify our approach to the problem at hand.
- **Magic.** We magically choose $x_0, x_1, \ldots, x_{n-1}$ as the special distinct x-values at which to evaluate $p(x)$.
- The matrix equation that represents the evaluation is:

Another Simplification

- **Set-up and notation.** For simplicity in upcoming arguments, we'll assume WLOG that n is a power of 2. (Why is this WLOG?)
- Matrix notation will greatly simplify our approach to the problem at hand.
- **Magic.** We magically choose $x_0, x_1, \ldots, x_{n-1}$ as the special distinct x-values at which to evaluate $p(x)$.
- The matrix equation that represents the evaluation is:
- $$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & & & \ddots & \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} p(x_0) \\ p(x_1) \\ \vdots \\ p(x_{n-1}) \end{bmatrix} \quad (*)$$

Choosing the magical x's

- **Next Question.** How can we cleverly choose $x_0, x_1, \ldots, x_{n-1}$ to reduce the computation time?

Choosing the magical x's

- **Next Question.** How can we cleverly choose $x_0, x_1, \ldots, x_{n-1}$ to reduce the computation time?

- Suppose we've chosen (magic again) $x_0, x_1, \ldots, x_{n-1}$ such that

Choosing the magical x's

- **Next Question.** How can we cleverly choose $x_0, x_1, \ldots, x_{n-1}$ to reduce the computation time?
- Suppose we've chosen (magic again) $x_0, x_1, \ldots, x_{n-1}$ such that
- $x_j = -x_{\frac{n}{2}+j}$ $\forall j = 0, 1, \ldots, \frac{n}{2} - 1$ (recall that n is a power of 2 and hence is even).

Choosing the magical x's

- **Next Question.** How can we cleverly choose $x_0, x_1, \ldots, x_{n-1}$ to reduce the computation time?
- Suppose we've chosen (magic again) $x_0, x_1, \ldots, x_{n-1}$ such that
- $x_j = -x_{\frac{n}{2}+j}$ $\forall j = 0, 1, \ldots, \frac{n}{2} - 1$ (recall that n is a power of 2 and hence is even).
- Then $x_0 = -x_{\frac{n}{2}}$, $x_1 = -x_{\frac{n}{2}+1}, \ldots, x_{\frac{n}{2}-1} = -x_{n-1}$.

Choosing the magical x's

- **Next Question.** How can we cleverly choose $x_0, x_1, \ldots, x_{n-1}$ to reduce the computation time?
- Suppose we've chosen (magic again) $x_0, x_1, \ldots, x_{n-1}$ such that
- $x_j = -x_{\frac{n}{2}+j}$ $\forall j = 0, 1, \ldots, \frac{n}{2} - 1$ (recall that n is a power of 2 and hence is even).
- Then $x_0 = -x_{\frac{n}{2}}$, $x_1 = -x_{\frac{n}{2}+1}, \ldots, x_{\frac{n}{2}-1} = -x_{n-1}$.
- We will use the above set of functional equations to reduce the problem to two subproblems.

Reduction to two subproblems

- Rewrite matrix equation (*) on the previous slide using the functional equations as shown next:

Reduction to two subproblems

- Rewrite matrix equation (*) on the previous slide using the functional equations as shown next:
- (**) $$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & & & \ddots & \\ 1 & x_{\frac{n}{2}-1} & x_{\frac{n}{2}-1}^2 & \cdots & x_{\frac{n}{2}-1}^{n-1} \\ & & & & \\ 1 & -x_0 & (-x_0^2) & \cdots & (-x_0^{n-1}) \\ 1 & -x_1 & (-x_1^2) & \cdots & (-x_1^{n-1}) \\ \vdots & & & \ddots & \\ 1 & (-x_{\frac{n}{2}-1}) & (-x_{\frac{n}{2}-1}^2) & \cdots & (-x_{\frac{n}{2}-1}^{n-1}) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{\frac{n}{2}-1} \\ \\ a_{\frac{n}{2}} \\ a_{\frac{n}{2}+1} \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} p(x_0) \\ p(x_1) \\ \vdots \\ p(x_{n-1}) \end{bmatrix}$$

Similarities in the red and black parts of (**)

- $$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & & & \ddots & \\ 1 & x_{\frac{n}{2}-1} & x_{\frac{n}{2}-1}^2 & \cdots & x_{\frac{n}{2}-1}^{n-1} \\ & & & & \\ 1 & -x_0 & (-x_0^2) & \cdots & (-x_0^{n-1}) \\ 1 & -x_1 & (-x_1^2) & \cdots & (-x_1^{n-1}) \\ \vdots & & & \ddots & \\ 1 & (-x_{\frac{n}{2}-1}) & (-x_{\frac{n}{2}-1}^2) & \cdots & (-x_{\frac{n}{2}-1}^{n-1}) \end{bmatrix}$$

Similarities in the red and black parts of (**)

- $$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & & & \ddots & \\ 1 & x_{\frac{n}{2}-1} & x_{\frac{n}{2}-1}^2 & \cdots & x_{\frac{n}{2}-1}^{n-1} \\ & & & & \\ 1 & -x_0 & (-x_0^2) & \cdots & (-x_0^{n-1}) \\ 1 & -x_1 & (-x_1^2) & \cdots & (-x_1^{n-1}) \\ \vdots & & & \ddots & \\ 1 & (-x_{\frac{n}{2}-1}) & (-x_{\frac{n}{2}-1}^2) & \cdots & (-x_{\frac{n}{2}-1}^{n-1}) \end{bmatrix}$$
- **Observation 1.** The coefficients of the even powers of x are the same in both the red and the black submatrices.

Similarities in the red and black parts of (**)

- $$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & & & \ddots & \\ 1 & x_{\frac{n}{2}-1} & x_{\frac{n}{2}-1}^2 & \cdots & x_{\frac{n}{2}-1}^{n-1} \\ & & & & \\ 1 & -x_0 & (-x_0^2) & \cdots & (-x_0^{n-1}) \\ 1 & -x_1 & (-x_1^2) & \cdots & (-x_1^{n-1}) \\ \vdots & & & \ddots & \\ 1 & (-x_{\frac{n}{2}-1}) & (-x_{\frac{n}{2}-1}^2) & \cdots & (-x_{\frac{n}{2}-1}^{n-1}) \end{bmatrix}$$
- **Observation 1.** The coefficients of the even powers of x are the same in both the red and the black submatrices.
- **Observation 2.** The coefficients of the odd powers of x in the red submatrix are the negatives of the corresponding odd powers of x in the black submatrix.

Red versus black; even powers versus odd powers of x

- **Notation.** Let $P(x) =$

$$\sum_{i=0}^{\frac{n}{2}-1} a_{2i}x^{2i} + \sum_{i=0}^{\frac{n}{2}-1} a_{2i+1}x^{2i+1}$$

Red versus black; even powers versus odd powers of x

- **Notation.** Let $P(x) =$

$$\sum_{i=0}^{\frac{n}{2}-1} a_{2i}x^{2i} + \sum_{i=0}^{\frac{n}{2}-1} a_{2i+1}x^{2i+1}$$

- The purple summation contains the even powers of x, and

Red versus black; even powers versus odd powers of x

- **Notation.** Let $P(x) =$

$$\sum_{i=0}^{\frac{n}{2}-1} a_{2i}x^{2i} + \sum_{i=0}^{\frac{n}{2}-1} a_{2i+1}x^{2i+1}$$

- The purple summation contains the even powers of x, and
- the orange summation contains the odd powers of x.

Red versus black; even powers versus odd powers of x

- **Notation.** Let $P(x) =$

$$\sum_{i=0}^{\frac{n}{2}-1} a_{2i}x^{2i} + \sum_{i=0}^{\frac{n}{2}-1} a_{2i+1}x^{2i+1}$$

- The purple summation contains the even powers of x, and
- the orange summation contains the odd powers of x.
- **More notation.** Let $P_e(x) = \sum_{i=0}^{\frac{n}{2}-1} a_{2i}x^i$, and

Red versus black; even powers versus odd powers of x

- **Notation.** Let $P(x) =$

$$\sum_{i=0}^{\frac{n}{2}-1} a_{2i}x^{2i} + \sum_{i=0}^{\frac{n}{2}-1} a_{2i+1}x^{2i+1}$$

- The purple summation contains the even powers of x, and
- the orange summation contains the odd powers of x.
- **More notation.** Let $P_e(x) = \sum_{i=0}^{\frac{n}{2}-1} a_{2i}x^i$, and
- let $P_o(x) = \sum_{i=0}^{\frac{n}{2}-1} a_{2i+1}x^i$.

$P(x)$ revisited

- $P_e(x) = \sum_{i=0}^{\frac{n}{2}-1} a_{2i}x^i$ and $P_o(x) = \sum_{i=0}^{\frac{n}{2}-1} a_{2i+1}x^i$.

$P(x)$ revisited

- $P_e(x) = \sum_{i=0}^{\frac{n}{2}-1} a_{2i}x^i$ and $P_o(x) = \sum_{i=0}^{\frac{n}{2}-1} a_{2i+1}x^i$.
- Then $P(x) = P_e(x^2) + xP_o(x^2)$. Verify!

$P(x)$ revisited

- $P_e(x) = \sum_{i=0}^{\frac{n}{2}-1} a_{2i}x^i$ and $P_o(x) = \sum_{i=0}^{\frac{n}{2}-1} a_{2i+1}x^i$.
- Then $P(x) = P_e(x^2) + xP_o(x^2)$. Verify!
- Moreover,
  $P(-x) = P_e((-x^2)) - xP_o((-x)^2) = P_e(x^2) - xP_o(x^2)$.

$P(x)$ revisited

- $P_e(x) = \sum_{i=0}^{\frac{n}{2}-1} a_{2i}x^i$ and $P_o(x) = \sum_{i=0}^{\frac{n}{2}-1} a_{2i+1}x^i$.
- Then $P(x) = P_e(x^2) + xP_o(x^2)$. Verify!
- Moreover,
  $P(-x) = P_e((-x^2)) - xP_o((-x)^2) = P_e(x^2) - xP_o(x^2)$.
- Thus to evaluate matrix (**), we've reduced the problem to evaluating $P_e(x^2)$ and $P_o(x^2)$ at $\frac{n}{2}$ points each:

$P(x)$ revisited

- $P_e(x) = \sum_{i=0}^{\frac{n}{2}-1} a_{2i}x^i$ and $P_o(x) = \sum_{i=0}^{\frac{n}{2}-1} a_{2i+1}x^i$.
- Then $P(x) = P_e(x^2) + xP_o(x^2)$. Verify!
- Moreover,
  $P(-x) = P_e((-x^2)) - xP_o((-x)^2) = P_e(x^2) - xP_o(x^2)$.
- Thus to evaluate matrix (**), we've reduced the problem to evaluating $P_e(x^2)$ and $P_o(x^2)$ at $\frac{n}{2}$ points each:
- The cost is $\frac{n}{2}$ additions, $\frac{n}{2}$ subtractions, and n multiplcations.

$P(x)$ revisited

- $P_e(x) = \sum_{i=0}^{\frac{n}{2}-1} a_{2i}x^i$ and $P_o(x) = \sum_{i=0}^{\frac{n}{2}-1} a_{2i+1}x^i$.
- Then $P(x) = P_e(x^2) + xP_o(x^2)$. Verify!
- Moreover,
  $P(-x) = P_e((-x^2)) - xP_o((-x)^2) = P_e(x^2) - xP_o(x^2)$.
- Thus to evaluate matrix (**), we've reduced the problem to evaluating $P_e(x^2)$ and $P_o(x^2)$ at $\frac{n}{2}$ points each:
- The cost is $\frac{n}{2}$ additions, $\frac{n}{2}$ subtractions, and n multiplcations.
- **So Far.** We now have two subproblems of size $\frac{n}{2}$ and $O(n)$ additional computations. Sound familiar?

$\frac{n}{2}$ and $O(n)$ additional computations...

- That is, if $T(n)$ is the run time of the algorithm, we are in the situation $T(n) = 2T(\frac{n}{2}) + O(n)$,

$\frac{n}{2}$ and $O(n)$ additional computations...

- That is, if $T(n)$ is the run time of the algorithm, we are in the situation $T(n) = 2T(\frac{n}{2}) + O(n)$,
- which is an $O(nlog(n))$-time algorithm.

$\frac{n}{2}$ and $O(n)$ additional computations...

- That is, if $T(n)$ is the run time of the algorithm, we are in the situation $T(n) = 2T(\frac{n}{2}) + O(n)$,
- which is an $O(nlog(n))$-time algorithm.
- Review Mergesort for more explanation.

$\frac{n}{2}$ and $O(n)$ additional computations...

- That is, if $T(n)$ is the run time of the algorithm, we are in the situation $T(n) = 2T(\frac{n}{2}) + O(n)$,
- which is an $O(nlog(n))$-time algorithm.
- Review Mergesort for more explanation.
- **The Point.** We've significantly reduced the complexity from the naive $O(n^2)$-time algorithm!!

How to find the magical x

- It remains to find the special values of x for which $x_j = -x_{\frac{n}{2}+j} \; \forall j = 0, 1, \ldots, \frac{n}{2} - 1$.

How to find the magical x

- It remains to find the special values of x for which $x_j = -x_{\frac{n}{2}+j}\ \forall j = 0, 1, \ldots, \frac{n}{2} - 1$.

- Let ω_n be a primitive nth root of unity. That is, $\omega_n = \cos(\frac{2\pi}{n}) + i\sin(\frac{2\pi}{n})$, where $i = \sqrt{-1}$.

How to find the magical x

- It remains to find the special values of x for which $x_j = -x_{\frac{n}{2}+j}\ \forall j = 0, 1, \ldots, \frac{n}{2} - 1$.

- Let ω_n be a primitive nth root of unity. That is, $\omega_n = \cos(\frac{2\pi}{n}) + i\sin(\frac{2\pi}{n})$, where $i = \sqrt{-1}$.

- In that case $\omega_n^n = 1$. For convenience, let $\omega_n = \omega$.

How to find the magical x

- It remains to find the special values of x for which $x_j = -x_{\frac{n}{2}+j} \ \forall j = 0, 1, \ldots, \frac{n}{2} - 1$.
- Let ω_n be a primitive nth root of unity. That is, $\omega_n = \cos(\frac{2\pi}{n}) + i \sin(\frac{2\pi}{n})$, where $i = \sqrt{-1}$.
- In that case $\omega_n^n = 1$. For convenience, let $\omega_n = \omega$.
- Geometrically, the complex numbers $\omega^0, \omega^1, \omega^2, \ldots, \omega^{n-1}$ are all vectors of length one spaced evenly around the unit circle centered at the origin of the complex plane.

How to find the magical x

- It remains to find the special values of x for which $x_j = -x_{\frac{n}{2}+j}\ \forall j = 0, 1, \dots, \frac{n}{2} - 1$.
- Let ω_n be a primitive nth root of unity. That is, $\omega_n = \cos(\frac{2\pi}{n}) + i\sin(\frac{2\pi}{n})$, where $i = \sqrt{-1}$.
- In that case $\omega_n^n = 1$. For convenience, let $\omega_n = \omega$.
- Geometrically, the complex numbers $\omega^0, \omega^1, \omega^2, \dots, \omega^{n-1}$ are all vectors of length one spaced evenly around the unit circle centered at the origin of the complex plane.
- The vector ω^1 has polar coordinates $(1, \frac{2\pi}{n})$ and to move on to the next vector on the list, we simply add $\frac{2\pi}{n}$ to the current angle: ω^2 has polar coordinates $(1, \frac{4\pi}{n})$, and so on.

Primitive roots of unity toolbox

- Let ω be a primitive nth root of unity. Then

Primitive roots of unity toolbox

- Let ω be a primitive nth root of unity. Then

  1. $\omega^n = 1$, and

Primitive roots of unity toolbox

- Let ω be a primitive nth root of unity. Then

    1. $\omega^n = 1$, and
    2. $\omega^0, \omega^1, \ldots, \omega^{n-1}$ are all distinct.

Primitive roots of unity toolbox

- Let ω be a primitive nth root of unity. Then
  1. $\omega^n = 1$, and
  2. $\omega^0, \omega^1, \ldots, \omega^{n-1}$ are all distinct.
- **Observation.** Every primitive nth root of unity has a multiplicative inverse since $\omega^k \omega^{n-k} = 1$.

Toolbox: Cancellation Property

- **Lemma.** If ω is a primitive n root of unity, then for each $k \neq 0$ with $-n < k < n$ we have

$$\sum_{j=0}^{n-1} \omega^{kj} = 0 \tag{1}$$

Toolbox: Cancellation Property

- **Lemma.** If ω is a primitive n root of unity, then for each $k \neq 0$ with $-n < k < n$ we have

$$\sum_{j=0}^{n-1} \omega^{kj} = 0 \tag{1}$$

- **Proof.** For any $k \neq 0$ with $-n < k < n$ we have $\omega^k \neq 1$ (why?) in which case (1) is a finite geometric series. Hooray!

Toolbox: Cancellation Property

- **Lemma.** If ω is a primitive n root of unity, then for each $k \neq 0$ with $-n < k < n$ we have

$$\sum_{j=0}^{n-1} \omega^{kj} = 0 \tag{1}$$

- **Proof.** For any $k \neq 0$ with $-n < k < n$ we have $\omega^k \neq 1$ (why?) in which case (1) is a finite geometric series. Hooray!
- Thus $\sum_{j=0}^{n-1} \omega^{kj} = \frac{(\omega^n)^k - 1}{\omega^k - 1} = \frac{1^k - 1}{\omega^k - 1} = \frac{0}{\omega^k - 1} = 0.$

Toolbox: Cancellation Property

- **Lemma.** If ω is a primitive n root of unity, then for each $k \neq 0$ with $-n < k < n$ we have

$$\sum_{j=0}^{n-1} \omega^{kj} = 0 \tag{1}$$

- **Proof.** For any $k \neq 0$ with $-n < k < n$ we have $\omega^k \neq 1$ (why?) in which case (1) is a finite geometric series. Hooray!
- Thus $\sum_{j=0}^{n-1} \omega^{kj} = \frac{(\omega^n)^k - 1}{\omega^k - 1} = \frac{1^k - 1}{\omega^k - 1} = \frac{0}{\omega^k - 1} = 0$.
- **QED**

Toolbox: Reduction Property

- **Lemma.** If ω is a primitive $2n$th root of unity, then ω^2 is a primitive nth root of unity.

Toolbox: Reduction Property

- **Lemma.** If ω is a primitive $2n$th root of unity, then ω^2 is a primitive nth root of unity.
- **Proof.** The complex numbers $1, \omega^1, \omega^2, \ldots, \omega^{2n-1}$ are all distinct (why?)

Toolbox: Reduction Property

- **Lemma.** If ω is a primitive $2n$th root of unity, then ω^2 is a primitive nth root of unity.
- **Proof.** The complex numbers $1, \omega^1, \omega^2, \ldots, \omega^{2n-1}$ are all distinct (why?)
- $\Rightarrow 1, \omega^2, \omega^4, \ldots, \omega^{2n-2}$ are all distinct.

Toolbox: Reduction Property

- **Lemma.** If ω is a primitive $2n$th root of unity, then ω^2 is a primitive nth root of unity.
- **Proof.** The complex numbers $1, \omega^1, \omega^2, \ldots, \omega^{2n-1}$ are all distinct (why?)
- $\Rightarrow 1, \omega^2, \omega^4, \ldots, \omega^{2n-2}$ are all distinct.
- Moreover, by definition, $\omega^{2n} = 1$, which means $(\omega^2)^n = 1$.

Toolbox: Reduction Property

- **Lemma.** If ω is a primitive $2n$th root of unity, then ω^2 is a primitive nth root of unity.
- **Proof.** The complex numbers $1, \omega^1, \omega^2, \ldots, \omega^{2n-1}$ are all distinct (why?)
- $\Rightarrow 1, \omega^2, \omega^4, \ldots, \omega^{2n-2}$ are all distinct.
- Moreover, by definition, $\omega^{2n} = 1$, which means $(\omega^2)^n = 1$.
- **QED**

Toolbox: Reflective Property

- **Lemma.** If ω is a primitive nth root of unity with n even then $\omega^{\frac{n}{2}} = -1$.

Toolbox: Reflective Property

- **Lemma.** If ω is a primitive nth root of unity with n even then $\omega^{\frac{n}{2}} = -1$.

- **Proof.** Use $k = \frac{n}{2}$ in the cancellation property:

Toolbox: Reflective Property

- **Lemma.** If ω is a primitive nth root of unity with n even then $\omega^{\frac{n}{2}} = -1$.
- **Proof.** Use $k = \frac{n}{2}$ in the cancellation property:
- $0 = \sum_{j=0}^{n-1} (\omega^{\frac{n}{2}})^j$

Toolbox: Reflective Property

- **Lemma.** If ω is a primitive nth root of unity with n even then $\omega^{\frac{n}{2}} = -1$.
- **Proof.** Use $k = \frac{n}{2}$ in the cancellation property:
- $0 = \sum_{j=0}^{n-1} (\omega^{\frac{n}{2}})^j$
- $= \omega^0 + \omega^{\frac{n}{2}} + \omega^{\frac{3n}{2}} + \cdots + \omega^{\frac{n}{2}(n-2)} + \omega^{\frac{n}{2}(n-1)}$

Toolbox: Reflective Property

- **Lemma.** If ω is a primitive nth root of unity with n even then $\omega^{\frac{n}{2}} = -1$.
- **Proof.** Use $k = \frac{n}{2}$ in the cancellation property:
- $0 = \sum_{j=0}^{n-1} (\omega^{\frac{n}{2}})^j$
- $= \omega^0 + \omega^{\frac{n}{2}} + \omega^{\frac{3n}{2}} + \cdots + \omega^{\frac{n}{2}(n-2)} + \omega^{\frac{n}{2}(n-1)}$
- $= \omega^0 + \omega^{\frac{n}{2}} + \omega^0 + \cdots + \omega^0 + \omega^{\frac{n}{2}}$

Toolbox: Reflective Property

- **Lemma.** If ω is a primitive nth root of unity with n even then $\omega^{\frac{n}{2}} = -1$.
- **Proof.** Use $k = \frac{n}{2}$ in the cancellation property:
- $0 = \sum_{j=0}^{n-1} (\omega^{\frac{n}{2}})^j$
- $= \omega^0 + \omega^{\frac{n}{2}} + \omega^{\frac{3n}{2}} + \cdots + \omega^{\frac{n}{2}(n-2)} + \omega^{\frac{n}{2}(n-1)}$
- $= \omega^0 + \omega^{\frac{n}{2}} + \omega^0 + \cdots + \omega^0 + \omega^{\frac{n}{2}}$
- $= \frac{n}{2}(1 + \omega^{\frac{n}{2}})$.

Toolbox: Reflective Property

- **Lemma.** If ω is a primitive nth root of unity with n even then $\omega^{\frac{n}{2}} = -1$.
- **Proof.** Use $k = \frac{n}{2}$ in the cancellation property:
- $0 = \sum_{j=0}^{n-1} (\omega^{\frac{n}{2}})^j$
- $= \omega^0 + \omega^{\frac{n}{2}} + \omega^{\frac{3n}{2}} + \cdots + \omega^{\frac{n}{2}(n-2)} + \omega^{\frac{n}{2}(n-1)}$
- $= \omega^0 + \omega^{\frac{n}{2}} + \omega^0 + \cdots + \omega^0 + \omega^{\frac{n}{2}}$
- $= \frac{n}{2}(1 + \omega^{\frac{n}{2}})$.
- Altogether, we have $1 + \omega^{\frac{n}{2}} = 0 \Rightarrow \omega^{\frac{n}{2}} = -1$. **QED**

Back to the magical xs

- Now let $x_j = \omega_j$ for $j = 0, 1, \ldots, n-1$.

Back to the magical xs

- Now let $x_j = \omega_j$ for $j = 0, 1, \ldots, n-1$.
- Observe that for each $j = 0, 1, \ldots, \frac{n}{2}$, we have $x_{j+\frac{n}{2}} = \omega^{j+\frac{n}{2}} = \omega^j \omega^{\frac{n}{2}} = -\omega^j = -x_j$.

Back to the magical xs

- Now let $x_j = \omega_j$ for $j = 0, 1, \ldots, n-1$.
- Observe that for each $j = 0, 1, \ldots, \frac{n}{2}$, we have $x_{j+\frac{n}{2}} = \omega^{j+\frac{n}{2}} = \omega^j \omega^{\frac{n}{2}} = -\omega^j = -x_j$.
- Thus the magical x's have been found!

Back to the magical xs

- Now let $x_j = \omega_j$ for $j = 0, 1, \ldots, n-1$.
- Observe that for each $j = 0, 1, \ldots, \frac{n}{2}$, we have $x_{j+\frac{n}{2}} = \omega^{j+\frac{n}{2}} = \omega^j \omega^{\frac{n}{2}} = -\omega^j = -x_j$.
- Thus the magical x's have been found!
- Finally note that in the subproblem of size $\frac{n}{2}$, the x-coordinates we choose will be $1, \omega, \omega^2, \omega^4, \ldots, \omega^{n-2}$.

Back to the magical xs

- Now let $x_j = \omega_j$ for $j = 0, 1, \ldots, n-1$.
- Observe that for each $j = 0, 1, \ldots, \frac{n}{2}$, we have $x_{j+\frac{n}{2}} = \omega^{j+\frac{n}{2}} = \omega^j \omega^{\frac{n}{2}} = -\omega^j = -x_j$.
- Thus the magical x's have been found!
- Finally note that in the subproblem of size $\frac{n}{2}$, the x-coordinates we choose will be $1, \omega, \omega^2, \omega^4, \ldots, \omega^{n-2}$.
- That is, substitute ω^2 for ω and repeat the procedure.

Conclusion and Summary

- **Conclusion.** The FFT accomplishes polynomial multiplication in $O(n \log n)$ time.

[1] `http://mathworld.wolfram.com/FourierTransform.html`

Conclusion and Summary

- **Conclusion.** The FFT accomplishes polynomial multiplication in $O(n \log n)$ time.
- **Summary.** Two Polynomials of degree $n-1 \rightarrow$ new representation in terms of points

[1] http://mathworld.wolfram.com/FourierTransform.html

Conclusion and Summary

- **Conclusion.** The FFT accomplishes polynomial multiplication in $O(n \log n)$ time.
- **Summary.** Two Polynomials of degree $n - 1 \rightarrow$ new representation in terms of points
- evaluate an arbitrary polynomial of degree $n - 1$ at n points

[1] http://mathworld.wolfram.com/FourierTransform.html

Conclusion and Summary

- **Conclusion.** The FFT accomplishes polynomial multiplication in $O(n \log n)$ time.
- **Summary.** Two Polynomials of degree $n-1 \rightarrow$ new representation in terms of points
- evaluate an arbitrary polynomial of degree $n-1$ at n points
- FFT accomplishes this in time $O(n \log n)$

[1] `http://mathworld.wolfram.com/FourierTransform.html`

Conclusion and Summary

- **Conclusion.** The FFT accomplishes polynomial multiplication in $O(n \log n)$ time.
- **Summary.** Two Polynomials of degree $n-1 \rightarrow$ new representation in terms of points
- evaluate an arbitrary polynomial of degree $n-1$ at n points
- FFT accomplishes this in time $O(n \log n)$
- recover coefficients of product polynomial with inverse FFT[1] in time $O(n \log n)$.

[1] http://mathworld.wolfram.com/FourierTransform.html

Next Week

NP-Completeness