Review

- Basic graph algorithms - Reachability - BFS/DFS/applications
- Greedy algorithms - Shortest paths, MST, clustering, Huffman codes, interval scheduling
- Divide and conquer

$\downarrow$

integer mult., matrix mult; DFT, closest pair, intro to parallelization, merge sort, polynomial mult.

DFT - review.

- Given $\vec{a} = (a_0, \ldots, a_{n-1})$; we want $DFT(\vec{a})$.

$$\begin{bmatrix} b_0 \\ \\ \vdots \\ \\ b_{n-1} \end{bmatrix} = \begin{array}{cc} & j \\ i & \begin{bmatrix} & & \\ & \omega^{ij} & \\ & & \end{bmatrix} \end{array} \begin{bmatrix} a_0 \\ \\ \vdots \\ \\ a_{n-1} \end{bmatrix}$$

$0 \le i,j \le n-1$.

ω is primitive n^{th} root of unity.

$$\omega^n = 1 \quad \text{and} \quad \forall k \in [1, n-1], \omega^k \neq 1.$$

Ex: (a_0, a_1)

ω is primitive 2^{nd} root of unity. | $\omega = -1$

$$\begin{bmatrix} w^{0\times 0} & w^{0\times 1} \\ w^{1\times 0} & w^{1\times 1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

$$DFT((a_0, a_1)) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} a_0 + a_1 \\ a_0 - a_1 \end{bmatrix}$$

Evaluation of p at

w^0, w^1.

$1, -1$

$P(x) = a_0 + a_1 x$.

$P(w^0) = P(1) = a_0 + a_1$

$P(w^1) = P(-1) = a_0 - a_1$

InvDFT is defined by A^{-1}.

$DFT = \text{Vand}(w^0, \dots, w^{n-1})$.

$$(A^{-1})_{i,j} = \frac{w^{-i \cdot j}}{n}$$

$\sum_{i=0}^{n-1} w^i = 0$

$\forall j$

$\sum_{i=0}^{n-1} w^{ij} = 0$

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -1/2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Vandermonde matrices

$$\text{Vand}(\alpha_0, \dots, \alpha_{n-1}) = \begin{bmatrix} \alpha_0^0 & \alpha_0^1 & \cdots & \alpha_0^{n-1} \\ \alpha_1^0 & \alpha_1^1 & \cdots & \alpha_1^{n-1} \\ \alpha_{n-1}^0 & \alpha_{n-1}^1 & \cdots & \alpha_{n-1}^{n-1} \end{bmatrix}$$

If $(\alpha_0, \alpha_1, \dots, \alpha_{n-1})$ are distinct points then this is a full rk matrix and inverses are easy to compute.

$\prod_{i \neq j} (\alpha_i - \alpha_j)$.

$$\begin{bmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix}\begin{bmatrix} a_0 + a_1 \\ a_0 - a_1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(a_0 + \cancel{a_1}) + \frac{1}{2}(a_0 - \cancel{a_1}) \\ \frac{1}{2}(\cancel{a_0} + a_1) - \frac{1}{2}(\cancel{a_0} - a_1) \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}$$

If $P(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$

In general,

$$\begin{bmatrix} P(w^0) \\ \vdots \\ P(w^{n-1}) \end{bmatrix} = \begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

Reed-Solomon codes.

$$\begin{tikzcd}
\overset{k}{\boxed{\text{message } m}} \arrow[r, "\text{Encode}"] & \overset{n}{\boxed{\text{Encoded message}}} \arrow[d] \\
\boxed{\text{message}' \; m'} & \boxed{\text{Channel/Storage}} \arrow[l, "\text{decode}"']
\end{tikzcd}$$

Want $m = m'$

$n \sim C \cdot k$

$d \leq \hat{n} - k + 1$

Singleton

$$\underbrace{(a_0, \ldots, a_{k-1})}_{\vec{a}} \xrightarrow{\text{Encoding}} (P_{\vec{a}}(\alpha_0), \ldots, P_{\vec{a}}(\alpha_{n-1}))$$

$\downarrow$ store it

$$(P_{\vec{a}}(\alpha_0), \ldots, \underset{\substack{\uparrow \\ \text{corruption introduced.}}}{\overset{P_{\vec{a}}(\alpha_i)}{\beta}}, \ldots P_{\vec{a}}(\alpha_{n-1}))$$

Berlekamp Welsh algorithm

$P(x) = a_0 + a_1 x + \cdots + a_{k-1} x^k$

If $f(x)$ and $g(x)$ are $\leq$ degree d polynomials, then $f(\alpha) = g(\alpha)$ for at most d many α's. $\longrightarrow$ Obs: $\geq d+1$ evaluations of f uniquely determine f

Interpolation: $f(x) = c_0 + c_1 x + \cdots + c_d x^d$

$\underbrace{f(\alpha_0), \ldots, f(\alpha_d)}_{\beta_0 \qquad \beta_d}$.

$\left\{ \sum_{i=0}^{d} c_i \alpha_j^i = \underset{\beta}{f(\alpha_j)} \right\}_{j=0}^{d}$