Everyone talks to teaching staff this week re project
(if you have not already) - esp. Singapore.

"Ideal" Parallel Computer (slide 2-3)
Problem: 
- # wires = \( \Theta(N^2) \) bad
- degree = \( \Theta(N) \) bad
- diameter = \( \Theta(1) \) good

Look at random routing or perm routing, else Helpful could make any
network look bad.

Desire: low-degree networks (slide 4)
- Linear array: \( \Theta(N) \) diameter
- 2D mesh: \( \Theta(\sqrt{N}) \) diameter
- Tree: \( \Theta(\log N) \) diameter

Thus Bounded degree \( \Rightarrow \Omega(\log N) \) diameter.

\[ \text{Dist} 0 1 2 3 k \quad \text{#nodes} 1 d d(d-1) d(d-1)^2 \Theta(d^k) \]

\[ \Theta(d^k) \geq N \Rightarrow k = \Omega(\log_d N) \]

Tree has low diameter, but is it a good routing network? No: congestion.

Def: Minimum bisection width = min # edges that
must be removed to partition network in half
(to within 1).

\[
\begin{align*}
\text{BW (tree)} &= 1 \\
\text{BW (array)} &= 1 \\
\text{BW (2D mesh)} &= \sqrt{N} \\
\text{BW (3D mesh)} &= N^{2/3}
\end{align*}
\]

Thus, \( N \) messages sent at random from \( N \) pro
\[ E[\text{Routing time}] = \Omega(\frac{N}{\text{BW} + \text{diameter}}) \]

Pf. Expect \( \Theta(N) \) messages to cross \( \text{BW} \) wire
- Both wire ships \( \leq 1 \) busy in unit time,
Time \( \geq \Theta(N)/\text{BW} \).
- Also Time \( \geq \text{diameter} \)
Hypercube (Slide 5-6)

Binary rep of node
\[ <b_{d-1}, b_{d-2}, \ldots, b_0> \]
connected to
\[ <\bar{b}_{d-1}, \bar{b}_{d-2}, \ldots, \bar{b}_0> \]
\[ <\bar{b}_{d-1}, \bar{b}_{d-2}, \ldots, \bar{b}_0> \]
\[ \vdots \]
\[ <\bar{b}_{d-1}, \bar{b}_{d-2}, \ldots, \bar{b}_0> \]

Two nodes connected if Hamming distance = 1.

Routing on hypercube:

10111010 \rightarrow 01101110

Flip any bit that's wrong by routing on that dimension. Bitwise XOR of current msg location and dest. Emit:

11010100 \rightarrow 00000000

Diameter = \( \lg N \) (\( \leq \))
Time = \( N (\lg N) \)
Degree = \( \lg N \)
BW = \( N/2 \), \( \Theta(N/\lg N) \) wires.

Cube-connected cycles (Slide 7)

\( N = n \lg n \) nodes.
Degree = \( \Theta(n) \) (\( \pm \), depending on whether wires are duplex)
Diameter = \( \Theta(n) \)
BW = \( \Theta(n) \) \( \leq \Theta(N/\lg N) \) since \( \lg N = \lg n + \lg \lg n \) = \( \Theta(\lg n) \)
**Butterfly (FFT) Network** (Slide 8-9)

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>n</td>
</tr>
</tbody>
</table>

(Direct network vs. indirect)

- Degree: \( \Theta(n) \)
- Diameter: \( \Theta(\log N) \) (little tricky if not 1 or 0)

\[ BW = \Theta(n) = \Theta(N/\log N) \]

Same as CCC, but authors didn't realize!

**Routing on butterfly** (Slide 21)

- Just like hypercube, but uses a specific order of dimensions

\[
\begin{align*}
\text{dest} & = 0 \Rightarrow \text{go up} \rightarrow \text{or} \rightarrow \text{go down} \rightarrow \text{or} \rightarrow \text{straight} \rightarrow \text{or} \rightarrow \text{cross} \\
\end{align*}
\]

- CBT rooted at each input (Slide 22)
- """" output (Slide 23)

**Decomposing a butterfly** (Slides 10-13)

- Remove "major cycles" \( \Rightarrow 2 \ n/2 \)-input butterflies
- Remove "minor" cycles \( \Rightarrow 2 \ n/2 \)-input butterflies (Slides 14-20)

**Packet routing**

- Source: \( X_{d_1} X_{d_2} \ldots X_0 \) \( \rightarrow \) \( Y_{d_1} Y_{d_2} \ldots Y_0 \)

Route major to minor:

- \( X_{d_1} X_{d_2} \ldots X_0 \)
- \( Y_{d_1} X_{d_2} \ldots X_0 \)
- \( Y_{d_1} Y_{d_2} \ldots X_0 \)
- \( Y_{d_1} Y_{d_2} \ldots Y_0 \)

\[ d = \log n \text{ steps} \]

But, might have congestion!
n packet on n-input butterfly.
What is worst-case perm?
\sqrt{n} packets at sources \(x_1, x_2, x_3, x_4, 0000\)
go to deets 0000 \(x_1, x_2, x_3, x_4\).
All go through 00000000 halfway through.
Network \(\Rightarrow\) congestion \(= \sqrt{n}\).

Benes network. (Slide 24-25)

Thm. Any n-perm can be routed (off-line) on
an n-input Benes with node-disjoint paths.
Pf. Induction on n.

Base \(N = 2\).

\[
\begin{array}{c|c}
2 & 1 \\
\hline
1 & 2
\end{array}
\]

Ind. case (Slides 26-35) \(\square\)

Corollary. An n-input Benes network can simulate
any n-node, degree-d network in \(O(d \log n)\) time \(\square\)

"But, butterfly is not so bad."

Theorem. Consider the \(N^N\) N-packet routing problems
on an N-node (\(n = \Theta(\log \log N)\)-input) butterfly.
At least \(N^N(1 - \gamma/n^2(\cdot))\) of these problems can
be routed in \(O(\log N)\) time.

Proof. We'll do a congestion-bound only that
will lead to an \(O(\log^2 N)\)-time result.
Phase 1: straight
WLOG, route packets to output, greedy input to output, let's...
Consider level-k node x during Phase 2.
# packets that can reach x = \(2^k \lg n\)
(true property, slide 23)
Prob. that given packet passes through node x \(\leq 2^{-k}\) (might not be able to reach x).
Consider any set of r specific packets.
Prob. they all pass through node x
\(\leq (2^{-k})^r = 2^{-kr}\) (independence)
Prob. that \(\geq r\) packets pass through node x
\(\leq (2^k \lg n)^r 2^{-kr}\)
\(\geq\) ways of choosing r packets

Note: This overcounts. If \(r+\Delta\) packets pass through x, this event will be counted \((r+\Delta)\) times within the \((2^k \lg n)^r\) ways.

\[\leq \left(\frac{e2^k \lg n}{r}\right)^r 2^{-kr}\]

\[\leq \left(\frac{ea}{b}\right)^b \text{ Deathbed}\]

Choose \(r = 2e \lg n\)
\[
\leq \left( \frac{1}{2} \right)^{2e\lceil N \rceil} \\
\leq N^{-2e} \\
\leq \frac{1}{N^{5.4}}
\]

Prob. that any node has \( \geq 2e \lceil N \rceil \) packets
\[
\leq N \cdot \left( \frac{1}{N^{5.4}} \right) \\
\text{\# packets}
\]
\[
\leq N^{-4.4}.
\]

\[\vdash N \left( 1 - \frac{1}{N^{4.4}} \right) \text{ problems see } \leq 2e \lceil N \rceil \text{ congestion.} \]

Hence, each level takes \( O(\lceil N \rceil) \) time \( \times \) \( \lceil N \rceil \) levels

\( = O(\lg N^2 N) \) time.

Phase 3 also takes \( O(\lg N) \) time, since \( O(\lg N) \) packets

\( \text{at each output.} \)

Corollary

\[ E[\text{routing time}] = O(\lg N) \cdot \left( 1 - \frac{1}{N^{4.4}} \right) + O(N) \cdot \frac{1}{N^{4.4}} \]

\[ = O(\lg N). \]