Exactly 2 weeks of class so

Final Presentations start Tuesday

25 minutes each

Presenting is hard to

very hard to just say what you're doing

What it is

Big results

Big ideas

Written paper 5-10 pages

Examples if you can

Signup in the wiki

3-connected
background
triangulations
canonical ordering
Schnyder realization
Realize to embedding
Optimizing embedding

Chernoff bound
OBLIVIOUS ROUTING in HYPERCUBES

(Valiant’s Algorithm)

Setup - computer network e.g. a cluster
with topology (underlying graph) is a hypercube

\[ \begin{array}{cccc}
\square & \square & \square & \square \\
\end{array} \]

Combintoially vertices are all \( 2^n \) bit strings
E edges are all pairs differing in one bit (Hamming dist. 1)

notice, \( n \)-regular, logarithmic \# edges, diameter \( n \), etc.

Routing problem
- one packet per node, \( P_i \)
- each packet has a destination \( \pi(i) \) (assume \( \pi \) a permutation)
- want to send \( P_i \) to node \( \pi(i) \)
- main constraint - each edge can send one packet per time step
- vertices can send packets to \( \pi \) at once

An OBLIVIOUS ROUTING ALGORITHM
where every node decides when to send packets independently
i.e. each step decided only on source, destination of packet

How long does it take?

LENGTH The LENGTH of a routing algorithm's schedule is \# of time steps before all packets arrive
$O(n)$ is the best we can hope for in, since
if diameter is $O(n^3)$.

Result - will be $O(8n)$.  

**First try** - Bit fixing (left-to-right routing)

Fix bits left to right

\[ \begin{array}{c}
000 \\
111 \\
011 \\
100 \\
110 \\
111
\end{array} \]

So and $p_{11} \implies 100 \implies 110 \implies 111$

This is oblivious, but has worst case behavior.

**Reason**

\[
\begin{array}{ccc}
\text{src} & \text{dest} & \text{flip} \\
0 & 0 & 0.00 \\
1 & 1 & 0.10 \\
2 & 2 & 0.11 \\
& & \text{flipped from src}
\end{array}
\]

Everyone piles up on 0!

\[ \Omega(2^{n/1}) \text{ packets arrive} \]

and $\Omega(2^{n/1})$ sets to "drain" (!)

Not too much harder to see any bit-fixing scheme will fail the same way.

Harder but not too bad (exercise)

Fixing bits in random order doesn't work either.

Ex. trick - do same input, look at expected number arriving at origin.

Get $\Omega(n)$ of some kind ($\Omega(n)$ cond.)
Valiant's algorithm

\textbf{input} \ p_i \ \pi(i)

Pick \ \sigma (circular permutation) uniformly randomly (expectations same if pick destinations randomly w/ replacement)

Rout the \ p_i to \ \sigma(i) using bid fixing
Rout \ \sigma(i) to \ \pi(i) using bid fixing

Will prove that w/ probability \( 1 - \frac{1}{N} \) or something, this completes in less than or equal to \( 8n \) steps

Analyze the delay of one packet in one step.

Define \ Fix a packet \ p_i. \ The path it takes is \ P.
\[ P = e_1, \ldots, e_k \]

\textbf{Def :} \( S = \{ \text{packets crossing } P \} \)
\[ = \{ p_j : \text{st. path } P(j) \text{ n } P \neq P \} \]

The \textbf{delay} of \( p_i \) is number of steps \( p_i \) waits.

Claim \ Delay(p_i) \leq |S|

If observation
once common subpath diverges, never reconverges, path never reconverges
account for all the delays of \( p_i \). When \( p_i \) gets delayed, it will put a token on the packet going on the edge it wants.
all packets pass coins along the path \( P \).

Claim - less than, or equal to \(|S|\) coins given out

No packet gets more than one coin, since coins move forward, don't pile up.
Now to estimate $|S|$

With probability $\Pr(|S| > 3n) \leq \frac{1}{n^3}$ from here will apply union bound, this implies average delay $4n$ since bounded since $n \to \infty$, $3n$ delay, $4n$ steps.

$$H_j = \begin{cases} 1 & \text{ if } p_j \in S \\ 0 & \text{ otherwise} \end{cases}$$

$$|S| = \sum_j H_j$$

Notice $H_j$ iid for fixed $P$ (bounded independent variables, Chernoff bounds apply to sum.)

Let $e \in P$

Def \( T(e) = \) # other packets crossing $e$

$$\sum_j H_j \leq \sum_e T(e)$$

strict because of "fellow traveling."

$$E(T(e)) \approx 2^{B-1} \frac{1}{2^B} = \frac{1}{2}$$