ANNOUNCEMENTS

➤ HW 3 will be out today; due Friday after fall break
➤ Look out for project topics! \[ \sim 20 \text{ papers} \]
➤ Groups of 2-3 \[ \leq 3 \text{ groups/paper} \]

(Sample project report)
TODAY’S PLAN

➤ Some applications of flows

➤ Randomness in algorithm design
Max flow: given a directed graph with capacities $c_e$ on the edges, and a source $s$, sink $t$, send as much flow as possible without violating capacities.

Min cut: given a directed graph with edge weights, find a small set of (weighted) edges to “cut” so that there is no longer a directed path from $s$ to $t$. 
EXAMPLE
WHAT WE SAW

➤ Max flow is always $\leq$ min cut (intuitively easy)

➤ Gives good way to certify optimality of flow algorithm

➤ Max-flow Min-cut theorem: in every graph, value of max flow is equal to the value of the min cut (thus certificate always exists)

➤ Proof algorithmic. gave algorithm for max flow; showed optimality by demonstrating a cut with value = value of flow — in any graph! (algorithm involved “residual graph” and “augmenting paths”)
Main idea: can we repair bad choices? (a la backtracking)

Algorithm: (initialize $f = 0$)

- Build residual $G_f$
- Find “augmenting path” $f'$ in $G_f$; set $f = f + f'$
- Repeat
THE BOTTOMLINE

➤ Max flow and min cut can be found in poly time

➤ Beautiful connection between them

➤ Many known algorithms for max flow — still active area of research (particularly directed graphs)
APPLICATIONS — CONNECTIVITY IN NETWORKS

- Vertex-disjoint paths

- "Edge-connectivity" — how many edges to remove so as to disconnect graph

- Undirected graph $G=(V, E)$

- Try all possible $s, t$
Know a few “foreground pixels” and “background pixels” — can we find the smallest cut? (edge wts correspond to similarity between neighboring pixels)

A cut gives a way of partitioning the entire image into foreground & background.
Assigning gifts to children — binary: suppose all happiness values are 0/1

Claim: this can be viewed as a flow problem.

Obs 0:
- every flow path has 3 edges.
- every child & gift is involved in at most one flow path.

What goes wrong when $H_{ij}$ are non-binary?
**APPLICATIONS — BASEBALL ELIMINATION**

Table 1: Standings of AL East on August 30, 1996.

<table>
<thead>
<tr>
<th>Team</th>
<th>Wins ($w_i$)</th>
<th>Losses ($l_i$)</th>
<th>To Play ($r_i$)</th>
<th>Games against each other</th>
</tr>
</thead>
<tbody>
<tr>
<td>NY</td>
<td>75</td>
<td>59</td>
<td>28</td>
<td>–  5  7  4  3</td>
</tr>
<tr>
<td>Baltimore</td>
<td>71</td>
<td>63</td>
<td>28</td>
<td>5  –  2  4  4</td>
</tr>
<tr>
<td>Boston</td>
<td>69</td>
<td>65</td>
<td>28</td>
<td>7  2  –  4  0</td>
</tr>
<tr>
<td>Toronto</td>
<td>63</td>
<td>71</td>
<td>28</td>
<td>4  4  4  –  0</td>
</tr>
<tr>
<td>Detroit</td>
<td>0  47</td>
<td>?</td>
<td>28</td>
<td>3  4  0  0  –</td>
</tr>
</tbody>
</table>

**Easy:** $w_5 = 46 \implies$ Detroit is eliminated

**What about $w_5 = 47$?**
RANDOMNESS IN ALGORITHM DESIGN

Motivating question: does randomness lead to faster algorithms?
TOY EXAMPLE

- A[1, ..., N] is an array in which N/2 elements promised to be zero. Find one i such that A[i] = 0.

- As long as you haven't found such i,
  - pick random i ∈ [N] \&
  - check if A[i] = 0.

- What is the prob. that it fails for t steps?
- prob of failure in each step ≤ \( \frac{1}{2} \).
- all steps ≤ \( \frac{1}{2^t} \).
- \( t \approx 20 \).
IDENTITY TESTING

Are two polynomials identically equal?

\[ p(x) = (x - 7)(x - 3)(x - 1)(x + 2)(2x + 5) \]
\[ q(x) = 2x^5 - 13x^4 - 21x^3 + 127x^2 + 121x - 210 \]

Say we pick a random \( x \) and check if \( p(x) = q(x) \) when \( x = 19 \).

When could it be that \( p(x) \neq q(x) \) but \( p(19) = q(19) \)?

\[ r(x) = p(x) - q(x) \]  
[Schwartz-Zippel Lemma]
Recall question: does a bipartite graph have perfect matching?

Introduce variable $x_{ij}$ for edge $e_{ij}$

$$M = \begin{bmatrix} x_{ij} & 0 \\ \end{bmatrix}$$

Fact: $\det(M) = 0$ if $M$ has no perfect matching

$\neq 0$ if $M$ has a perfect matching.
\[ M = \begin{bmatrix} x & y & 0 \\ 0 & 0 & w \\ 0 & z & 0 \end{bmatrix} \]

\[ \det(M) = \sum_{\text{permutations}} (-1)^{\text{sign}(\sigma)} x^{\sigma(1)} w^{\sigma(2)} z^{\sigma(3)} \]

**Obs 1:** we have one term per \textit{one} \textit{to} \textit{two} \textit{to} \textit{three} \textit{to} \textit{four} \textit{to} \textit{five} \textit{to} \textit{six} \textit{to} \textit{seven} \textit{to} \textit{eight} \textit{to} \textit{n} \textit{to} \textit{per} \textit{mutation}.

**Obs 2:** given values for \textit{x}, \textit{y}, \textit{z}, \textit{w} we can evaluate \[ \det(M) \] in \textit{polynomial} \textit{time}, \textit{(Gaussian elim.)
Recall question: does a bipartite graph have perfect matching?
PRIMALITY TESTING (MILLER–RABIN TEST)

- Is a given number $q$ prime?

- Test($a, q$): involves exponentiating $a$, ... (all doable in $\text{poly(#digits)}$)

- **Key property:** if $q$ is prime, test passes for all $a < q$, and if not, test fails for half the numbers $a < q$!
**Primality Testing:**

\[ q = (q_1, q_2 \ldots q_n) \]

**Standard Test:**

- for \( q \) check all \( d \) : \( 2 \ldots \sqrt{q} \)
  - if \( d \mid q \) is time \( \text{poly}(n) \)

\[ \frac{n}{2} \text{ \text{poly}(n)} \rightarrow \text{exp.} \]

**60's:**
- If you allow randomized ALGs, can be \([\text{Miller-Rabin}]\) done in \( \text{poly}(n) \) time.

**[2002]**
- Agarwal, Kayal, Saxena: A deterministic \( \text{poly}(n) \) time for this.
GUARANTEES FOR RANDOMIZED ALGORITHMS

➤ Running time can be varying — expected running time, ...

➤ Success probability

➤ General tradeoff — boosting