• **DISCLAIMER:** These notes are not necessarily an accurate representation of what I said during the class. They are mostly what I intend to say, and have not been carefully edited.

• List ranking
  
  – Linked list, each node knows predecessor / successor, and has value.
  
  – Goal: calculate prefix sum.
  
  – Application: place items in an array, in parallel.
  
  – Simple algorithm: walk from head of list, \(O(n)\) steps.

• Parallel Complexity
  
  – Model: each node can store extra information, point to other nodes.
  
  – Note: this is an approximation of parallel machine.
    
    * Many other models: distributed, Mapreduce, streaming.
    
    * Will attempt to be ‘model independent’.
  
  – Goal for list ranking: parallel complexity, \(O(\log n)\) time, \(O(n)\) work.

• Pointer jumping
  
  – Every node looks at predecessor’s predecessor, set that as new predecessor.
    
    * Jumps twice as long each step
    
    * \(O(\log n)\) rounds, each \(O(n)\) work, total \(O(n \log n)\).
  
  – Every second node merges with predecessor.
  
  – list half as big, recurse on it.
  
  – Hierarchy of lists, each half as big as the other: \(O(\log n)\) rounds, \(O(n)\) total work.
  
  – Difficulty: finding every other pointer: this is global information.

• Administration
  
  – This course in one sentence: design algorithms of this form for graphs, matrices, and things in between.
- Enrollment: can enroll more, but need different evaluation scheme.
- Evaluation: problems, solve any 10, present to me or hand in solutions.
- Presentations / projects: depend on enrollment.
- ??? (separate) coding component ???

- Finding every other point
  - Alternate view of pairing: odd locations ‘receive’, even locations ‘send’.
  - Randomized solution: nodes declare themselves as odd/even w.p. 1/2.
  - Issue: even sending to an even. Fix: do nothing on both.

- What if all nodes declare themselves even?
  - $X_e$: edge $e$ disappears this step: $\mathbb{E}[X_e] = 1/4$.
  - $Y$: random variable for remaining size.
  - $Y \geq 0, \mathbb{E}[Y] = 3/4n$.
  - Markov inequality:
    \[
    \Pr [Y \geq t] \leq \frac{\mathbb{E}[Y]}{t}.
    \]
  - Apply with $t = 9/10n$: $\Pr [Y \geq 9/10n] \leq (3/4)/(9/10) = 5/6$.
  - W.h.p. $O(\log n)$ steps until size decreases by a constant factor, $O(\log^2 n)$ total rounds.
  - Can also use concentration bounds on all iterations.

- Remarks
  - Prototypical V-cycle algorithm.
  - Can also view as top down divide-and-conquer.
  - Analogous to how treap (a form of balanced BST) works: sampling and data structures have close connections.