• **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.

• Main topics:
  - Homework 1 returned: mean 11.6/15, median 13.0/15.
  - Binary trees revisited: they are just divide-and-conquer.
  - Static construction, offline.
  - Augmentation: can store statistics about entire subtrees.

• Comments from last time
  - Lots of undefined symbols / terminologies.
  - Review session / notes? Most important are things that’s on homework 1.
  - How to choose polynomial? There are only two possibilities that we can address: line up the sequences forwards / backwards. Can then further narrow down with example / checking coefficients.

• Test1: in class this Friday (Sep 7, 2016)
  - So far no requests to take it early / late.
  - You may use a sheet of notes on both sides.
  - Coverage: lectures 1-6.
  - Main Topics
    * Asymptotic complexity: $O$, $\Omega$, and $\Theta$.
    * Designing divide-and-conquer algorithms.
    * Setting up runtime recurrences.
    * Solving recurrences using Master theorem (other methods are optional).
    * Applications of fast multiplication.
  - NOT included:
    * Definition and algorithm of inversion counting.
    * Details of how to multiply numbers faster than $n^2$.  

* Guess and check / recursion tree: master theorem works for everything on
  test.

• Binary Trees
  - Rooted tree, each node has a key.
  - Left subtree of $x$: all with keys smaller than $x$, right subtree: all with keys
    larger than $x$.
  - Goal: tree depth $O(\log n)$.

• Balancing:
  - Unbalanced case: long path.
  - Fix: AVL, Redblack, etc etc etc.
  - Simpler fix: if we know all the keys that we’ll ever insert, pick root to be
    median of all keys beforehand.
  - Depth: same as binary search, $\log_2 n + 1$.

• Insert/delete?
  - Build entire tree.
  - Flag nodes as ‘present’ or not.
  - $O(\log n)$ time rank and update by storing subtree information.
  - Simplification: internal nodes no longer carry keys, just store ‘merged’ infor-
    mation states of their descendants.