Outline for June 1, 2009

Reading: text, §13.3  
Guest Lecturer: Justin Cummins

1. Recursive refresher: Fibonacci sequence
2. Merge function
   a. Merges two sorted lists into a third sorted list
      i. Compare first item of each list
      ii. Return the smaller value concatenated with the merge of the two lists
          (minus the smaller value)
      iii. If one list is empty, the result is the other list (terminating case/condition)
   b. Linear time $O(n)$: each compare places one item into the result, where the result
      has $n$ items
3. Mergesort function (see mergesort.py)
   a. One of the fastest general-purpose sorts
   b. Divide and Conquer algorithm
   c. Parallelizes easily
   d. Python’s list sort is a modified mergesort
   e. Algorithm
      i. Divide list in half
      ii. Recursively sort each half
      iii. Merge the sorted parts together
      iv. If a list has 0 or 1 element, it’s already sorted (terminating case/condition)
   f. Example: Mergesort class volunteers by height
   g. $O(n \log n)$ algorithm in worst case and average case
4. Compare mergesort to selection sort on random list of integers
   a. Selection sort is a quadratic, $O(n^2)$, algorithm
   b. For 10 and 100 elements, selection sort slightly faster
   c. For 1000 elements, mergesort almost 10× faster
   d. For 10000 elements, mergesort over 100× faster