Computer Science 134C

Introduction to Computer Science, in Python

Lecture #24 (Recursive sorting)

November 9

Keywords
mergesort,partition,quicksort

We focus on implementing two recursive sorts.

1. Questions?
2. Some reminders about logarithms.
   (a) Logarithms are defined by the relationship $n = b^{\log_b n}$.
   (b) The value $\log_b n$ is, essentially, the number of times n must be divided by b to reduce it to 1.
   (c) In practice, we often use $b = 2$ because we often develop problem solutions that involve dividing the size of inputs in half.
   (d) Since $\log_a n = \log_a b \cdot \log_b n = c \log_b n$ we can see that logarithms of various bases differ by a constant multiplier.
   (e) Computer scientists write $O(\log n)$ to describe logarithmic function growth, no matter the base.
3. Quicksort.
   (a) Notice that we can identify where any value is correctly located within the sorted list, without actually sorting the list. For example, the smallest value should end up in location 0. We'll use a technique, called *partitioning* that finds the appropriate place for a *pivot* by putting smaller values to its left and larger values to its right.
   (b) Once partitioned, we can sort (somehow!) the smaller and larger values independently.
   (c) This potentially is a fast sort, unless your data is in-order, or in reverse-order. Why?
   (d) A fix: pick a pivot randomly.
4. Merge sort.
   (a) Split a list into two sublists; sort the sublists; merge them.
   (b) The merge operation: from two sorted lists construct a new sorted list containing all the values.

⋆