Lab 3
Due: at the end of class

1 Reading

Skim Chapters 7.1 – 7.3 of the textbook.

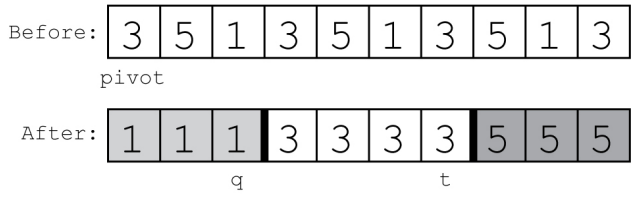
2 Exercises

Write out solutions to the exercises below, following the format in the textbook. Please include the names of everyone in your group.

1. The lecture's version of quicksort performs poorly on already-sorted arrays. Consider the book's partition algorithm which always picks the rightmost index in a partition as the pivot, rather than the leftmost. Show the parameters (A, l, r) for every partition call required to run the modified quicksort on $A = [1, 3, 5, 6, 10]$. Draw the updated array after each partition call, with the current range labeled. Is this more efficient than running the leftmost-pivot version?

2. The PARTITION(A, l, r) function picks a pivot p and returns the value i where elements $A[l..i] \leq p$ and elements $A[i+1..j] > p$. If all of the values in A are distinct, this gives us three tidy partitions. If there are lots of repeated values in A, the left partition may end up oversized because of many values that are equal to p. Write a modified version of the PARTITION pseudocode that splits the array into three partitions: elements in $A[l..q] < p$, elements in $A[q+1..t] = p$, and elements in $A[t+1..r] > p$. The function should return **two** values: q and t. (Never mind that Java can't return two values from a function. This is pseudocode so we can do whatever we want).

   Run through an example of your pseudocode on scrap paper to convince yourself that it works correctly. Only turn in the pseudocode.



3 Grading

Exercise 1: 40%

Exercise 2: 60%