

Programming Assignment #2

Due on 04/03/2014

1. Please implement a Quicksort in C or C++ and compare running time with following two Quicksorts
2. (Median-of-3 Partition) One way to improve the **RANDOMIZED-QUICKSORT** is to choose the pivot for partitioning more carefully than by picking a random element from the array. One common approach is to choose the pivot as the median of a set of 3 elements randomly selected from the array. Assume that all elements in the array are distinct. Please implement a Quicksort with Median-of-3 Partition in C or C++ and answer following questions.
 - (a) (review) What is the probability of getting an OK split if the pivot is chosen at random? Explain. (A split is “OK” if the smaller piece has at least $n/4$ elements.)
 - (b) Roughly, what is the probability of getting an OK split with the new median-of-3 method? Explain.
 - (c) Let I be the indicator random variable for getting an OK split using the median-of-3 partition:

$$I = \begin{cases} 1 & \text{if the split is OK} \\ 0 & \text{otherwise} \end{cases}$$

What is the expectation of I ?

3. Bentley and McIlroy describe an implementation of Quicksort based on a new pivoting-finding algorithm by using a partition on Tukey's ninther which is median of the median of 3 samples and each sample has 3 elements. The following figure illustrates the Tukey's ninther. Please implement the Bentley-McIlroy Quicksort algorithm in C or C++.

[illegible]