Foundations of Algorithms
Programming Assignment #2
OR
Just When You Thought You Were Done With Sorting....

The following is a problem to be completed by the individual (i.e., it is not collaborative) and then implemented. Please follow the requirements provided under Syllabus & Course Information under the link Programming Assignment Guidelines. Pseudocode must be in the style of the text and conform to the Pseudocode Conventions on pages 20-22 of the text.

Assignment Tips:

- Follow the Programming Assignment Guidelines to submit source code and analysis.
- Don't forget to include a rigorous analysis component to your submission!
- Remember that CPU time is not a valid measure for testing run time. You must use something such as the number of comparisons.

Assignment Details

Intro to Introsort

The **Introsort** algorithm is a hybrid sorting algorithm that attempts to combine the benefits of quicksort and heapsort. In this assignment, you will implement and analyze this algorithm. Feel free to use online resources that describe the algorithm, short of duplicating the source code itself.

1. [10 points] Write pseudocode for Introsort, Quicksort, and Heapsort (a required subfunction of Introsort), and analyze the worst-case runtimes of all three algorithms.
2. [40 points] Implement Quicksort, Heapsort, and Introsort.
3. [20 points] For a range of array sizes (e.g., 10, 100, 1000,...), compare the empirical runtime of the three sorting algorithms. Do this comparison for input arrays that are reverse-sorted (i.e., the Quicksort worst case) and randomized. For this portion of the assignment, assume that the partition element is chosen as the final element of the array.

Median-of-Three Partitioning

Given an array , $a[i], ..., a[j]$, with $j - i \geq 2$, let $k = \lfloor (i + j)/2 \rfloor$ and suppose that you chose as the partition element the median of $a[i]$, $a[j]$, and $a[k]$ (i.e., the value that would be in the middle if $a[i]$, $a[j]$, and $a[k]$ were sorted). This is called median-of-three partitioning.

1. [5 points] Write pseudocode for using median-of-three in the Quicksort and Introsort *Partition* function.
2. [5 points] What is the running time of median-of-three partitioning? Justify your answer.
3. [20 points] Repeat your previous runtime experiments (described above) using the median-of-three partitioning. How do these empirical runtimes compare to your previous results?