1 Assignment 2

Submit source code and running instructions to EAS[1]. Do it in Java. Do not use java's search methods! Do not use Java's Collections or Subclasses, code your own. Place textual responses for 2, 3 and 4 in block comments in your code.

Posted: Thursday, May 18^{th}

Due: Monday, May 29^{th}

Grade: 5%

1. Quicksort
   (a) Implement a generic Quicksort algorithm that takes an array as input, it should use trivial pivot selection.
      i. This file should be called QSNormal.java
      ii. This class should have a sort method:
      void sort(int[] input)
   (b) Implement a Quicksort algorithm that uses a Median of Three pivot selection.
      i. This file should be called QSMedian.java
      ii. The class should have a sort method:
      void sort(int[] input)
   (c) Write classes that generate test inputs of size 10, 100, 10000, 1000000.
      i. One file should be RandomGen.java
      ii. One file should be FixedGen.java
      iii. RandomGen should generate uniformly random integers.
      iv. FixedGen should always generate a fixed input.
   (d) Make a driver that sorts values from your input
      i. This file should be called QSDriver.java
      ii. This file should output the run-time in either ns or μs
      iii. it should accept command-line as follows:
      java QSDriver <sort> <gen> <length> <seed>
         A. <sort> is either QSNormal or QSMedian
         B. <gen> is either RandomGen or FixedGen
         C. <length> is the number of ints to be sorted in the input array
         D. <seed> is an optional argument that lets you repeat the random seed for RandomGen (but is ignored by FixedGen)

[1] https://fis.encs.concordia.ca/eas/

(e) Record performance times of runs for each input size specified in 1c for Quicksorts implemented in 1a and 1b using RandomGen.

2. In clear, natural language, describe the performance differences between the two sorts. Try to correlate this with the underlying mechanism.

   (a) This textual response should be no more than 8 lines / 80 words.

3. In clear, natural language, describe a pathological input (one that yields a worst-case) for your trivial Quicksort defined in 1a. The FixedGen.java file should produce this pathological input.

   (a) This textual response should be no more than 8 lines / 80 words.

4. In clear, natural language, describe how the pathological case performs given the Median of Three Quicksort. Reference performance tests that you run and your understanding of what makes that input a pathological case. 1a.

   (a) This textual response should be no more than 10 lines / 100 words.