

Project #2 – Sort Race

Introduction

This project is to write an Html+Javascript program for a racing competition between different sorting algorithms. Your program will run several algorithms in an interleaved fashion, where each algorithm will get a turn to perform a single step (one of its complexity operations) in its run, and maybe rearrange its array.

This project will involve a Race Mgr that will loop, giving each algorithm a turn on each iteration by calling its pass() function. Finally, the Race Mgr will stop calling an algorithm's step() function when it returns that it has completed its sorting task. Each algorithm should be given its own copy of the array of integers to be sorted.

Algorithms

For this project, you will run three sorting algorithms: the $O(N^2)$ Insertion sort, the $O(N \log N)$ Merge sort, and the $O(N \log N)$ Quicksort.

Setup

Your program should select one of the sample inputs provided below at random. Each is a list of 12 hex digits. You will create an array of these integers.

Race Mgr & Step() Methods

The Race Mgr should loop until the first sort algorithm reports that it is done. (It should finish the iteration in which an algorithm reports done.) Each iteration should call the Step() function for each algorithm. The Step() function should run the algorithm through one more complexity operation, which is usually one comparison between elements and what the algorithm does based on the result of that comparison. The Step() function should also call a UI display update function to allow the display to update that algorithm's part of the display, if needed. This will allow the user to see each algorithm improve its position in the race.

Display

You should display three races each as a row array in its own vertical grid starting at the top row with the raw input elements. The row-by-row display is similar to the cellular automata display, but in 3 columns, one for each algorithm. Assume the input list to sort is only 12 elements comprised of hexadecimal digits taken from the range 0..B. Below are sample inputs that your program should be able to race. For a race, each sorting algorithm will start with the same input. You should be able to run any of the sample inputs. A new row for an algorithm should be provided as each “pass” is completed. While the steps/comparisons of an algorithm are being done, changes in element positions should be displayed. The prior row for the prior “pass” should be left as is, so that the user can see the “pass” differences.

Sample Inputs

(0, B, A, 3, 2, 8, 4, 7, 6, 5, 1, 9)	(6, 9, 8, 7, 2, B, 3, A, 5, 4, 1, 0)
(0, A, 9, 8, 1, A, 3, 9, 2, 0, 1, 1)	(6, A, 2, 3, 0, 5, 3, 0, 4, 7, 8, 1)
(1, 6, 3, 8, 9, 4, 0, A, 5, 2, B, 7)	(7, 0, 1, A, 6, 9, 3, 5, 4, 2, B, 8)
(1, 9, 8, 4, 1, B, 3, 8, 2, 6, 2, 5)	(7, 8, 5, 2, 8, 6, 1, 0, 3, 4, 2, 9)
(2, 9, 7, B, 4, 0, 1, 6, 3, 8, A, 5)	(8, 7, 3, A, 9, 4, 2, 5, B, 1, 6, 0)
(2, 6, 1, 0, 9, 4, 8, 7, 8, 6, 2, 6)	(8, A, 1, 5, 9, 3, 4, 7, 9, 0, 8, 5)
(3, 4, 5, 7, 1, 9, 2, 0, 6, 8, B, A)	(9, 0, B, 3, 4, 2, 7, 5, 6, 1, 8, A)
(3, 5, 6, A, A, 0, 2, 3, B, 7, 2, 4)	(9, 9, B, 5, 3, 5, 1, A, 3, 3, A, B)
(4, B, 6, 0, 7, 9, A, 2, 1, 8, 3, 5)	(A, 4, 0, B, 5, 8, 6, 1, 7, 9, 2, 3)
(4, 0, B, 0, 6, 5, 6, 6, 7, 1, 0, A)	(A, 3, 9, 5, 9, A, 2, 2, A, 4, 4, 4)
(5, 2, 8, 1, A, B, 3, 4, 7, 9, 0, 6)	(B, 8, A, 4, 6, 3, 7, 9, 0, 1, 5, 2)
(5, 7, 5, 0, 6, 8, 4, B, 8, 9, 3, 4)	(B, 6, 0, 0, 5, A, 6, 2, 7, B, 2, 3)

335 — Algorithm Engineering — Sort Race

Complexity Order

You should prepare a 1-page (at most) paper describing your analysis of the Big-O running time of whatever algorithm you use. Address the usual issues such as main operations, input size, etc.

Team

The team size is the same as before, but you can change team members from the previous project if you wish.

Project Reports

As before, but simplified a bit. Please check out the updated sample Report pdf.

Readme File

As before.

Academic Rules

Correctly and properly attribute all third party material and references, if any, lest points be taken off.

Submission

As before.

Grading

As before.