Report- HW2 Cüneyt EREM, sec-1, 21202398

We write the console this line and compile it, then write ‘a’ to see the results

g++ -std=c++11 main.cpp findMedian.cpp findMedian.h then enter ‘a’

For example this is for the random array’s output result;

****

Arrays are created with size = 1000

Algorithm-1 Execution took 0 milliseconds.

Algorithm-2 Execution took 0 milliseconds.

Algorithm-3 Execution took 0 milliseconds.

****

Arrays are created with size = 2000

Algorithm-1 Execution took 15.62 milliseconds.

Algorithm-2 Execution took 0 milliseconds.

Algorithm-3 Execution took 0 milliseconds.

****

Arrays are created with size = 4000

Algorithm-1 Execution took 15.625 milliseconds.

Algorithm-2 Execution took 0 milliseconds.

Algorithm-3 Execution took 0 milliseconds.

****

Arrays are created with size = 8000

Algorithm-1 Execution took 62.504 milliseconds.

Algorithm-2 Execution took 0 milliseconds.

Algorithm-3 Execution took 0 milliseconds.

****

Arrays are created with size = 16000

Algorithm-1 Execution took 234.377 milliseconds.

Algorithm-2 Execution took 0 milliseconds.

Algorithm-3 Execution took 0 milliseconds.

****

Arrays are created with size = 32000

Algorithm-1 Execution took 930.348 milliseconds.

Algorithm-2 Execution took 0 milliseconds.

Algorithm-3 Execution took 0 milliseconds.

****

Arrays are created with size = 64000

Algorithm-1 Execution took 3781.5 milliseconds.

Algorithm-2 Execution took 15.643 milliseconds.

Algorithm-3 Execution took 0 milliseconds.

****

Arrays are created with size = 128000

Algorithm-1 Execution took 14971.8 milliseconds.

Algorithm-2 Execution took 15.626 milliseconds.

Algorithm-3 Execution took 15.637 milliseconds.

****

Arrays are created with size = 256000

Algorithm-1 Execution took 59943 milliseconds.

Algorithm-2 Execution took 46.877 milliseconds.

Algorithm-3 Execution took 31.253 milliseconds.

****

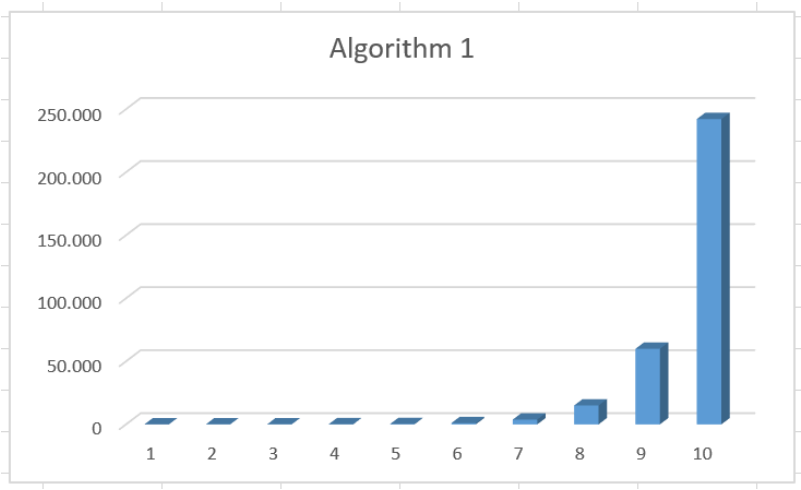
Arrays are created with size = 512000

Algorithm-1 Execution took 242128 milliseconds.

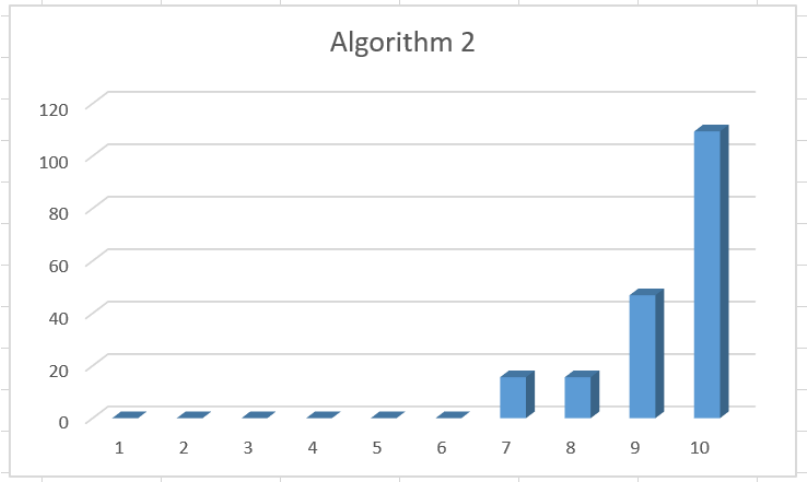
Algorithm-2 Execution took 109.371 milliseconds.

Algorithm-3 Execution took 46.881 milliseconds.

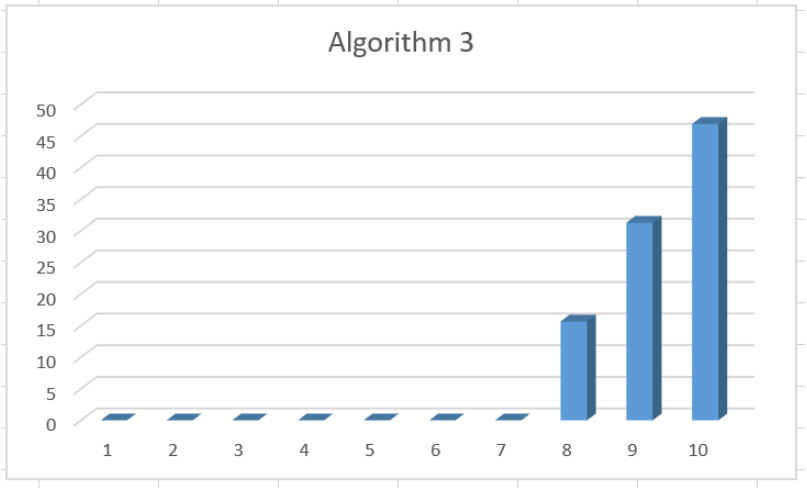
Tables that running time X array input size (1 to 10 means size 1000 to 512000) for 3 different algo;



O(n^2)



O(nlogn)



O(n)

We have array inputs as 1000, 2000 … to 512000 so that have different time elapses, also same input array is used for 3 different algorithms. Computer has i7-4700 hq 2.4 ghz processor and 12 gb ram and has Windows 10 operating system. When we look at the variables and also tables experimental results as follows, for the first algorithm, line is rising aproximately 4 times in each step, meaning n^2 or O(n^2) quadratic rate. It look like theoritical plot rate but not exactly the same for example array input size = 8000-16000 rate is 234.37/62.5 = 3.75 but for input size = 16000-32000 rate is 930,34/234.37 = 3,96 so that rate is close to rate 4.00 but not excatly the same. Similar rate differences are in other algorithm 2 and 3 so that theoritical and experimental differences are only minor. Second algorithm rises at n*logn rate. Third algorithm rises at linear rate O(n). So, experimental results and theoretical result are very close to each other and as we expected.