Complexity Experiment

Range algorithms
comparison report

By:

MHD Maher Azkoul
438017578
s438017578@st.uqu.edu.sa
course: Data Structures Course

Professor:
pr. Mohammad Nour

Umm Al-Qura University
College of Computers and Information Systems
Data Structures Course – 14012401 - 1441

Oct 19, 2019

Github Link

Content:

Objectives
Algorithms
Results
Conclusion

Objectives:

This is a report about comparing the time complexity of three algorithms that find the range for a given array of numbers.
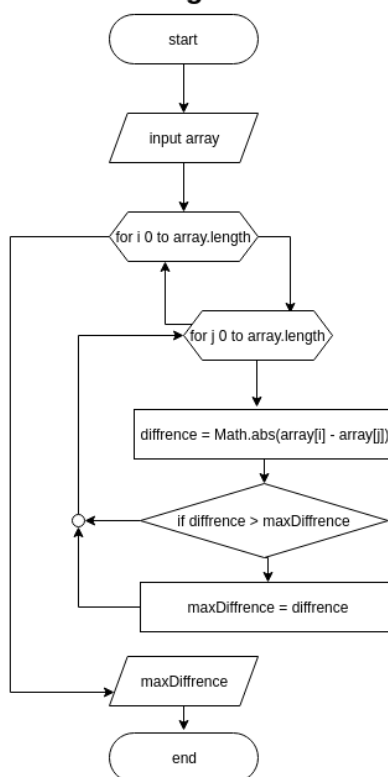
First algorithm is order of $O(N^2)$. Second Algorithm is order of $O(N^2)$, but it takes half of time that first algorithm takes. Third Algorithm is order of $O(N)$.
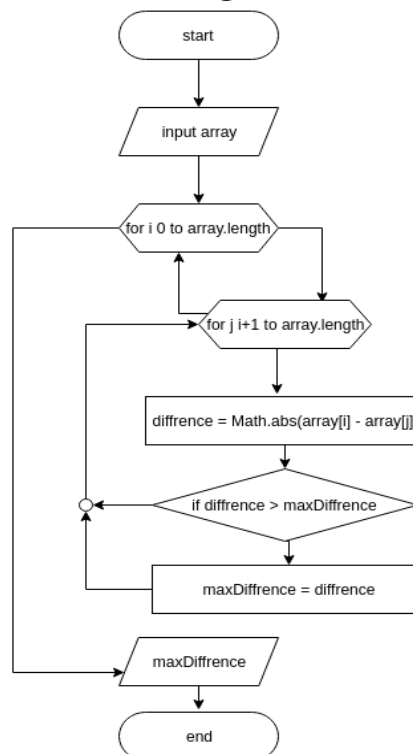
Environment & Tools:
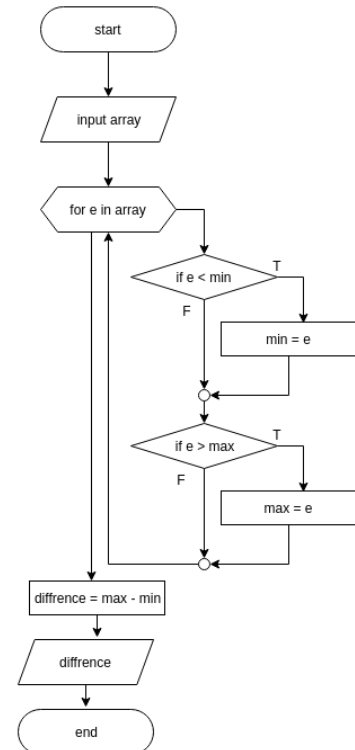
This experiment is written in java.

Algorithms:

First Algorithm

start

input array

for i 0 to array.length

for j 0 to array.length

diffrence = Math.abs(array[i] - array[j])

if diffrence > maxDiffrence

maxDiffrence = diffrence

maxDiffrence

end

Second Algorithm

start

input array

for i 0 to array.length

for j i+1 to array.length

diffrence = Math.abs(array[i] - array[j])

if diffrence > maxDiffrence

maxDiffrence = diffrence

maxDiffrence

end

Third Algorithm

start

input array

for e in array

if e < min   T

F

min = e

if e > max   T

F

max = e

diffrence = max - min

diffrence

end

2

Umm Al-Qura University   **Student: MHD Maher Azkuol**
Data Structures – 1441   **ID: 438017578**
Complexity Experiment   **Pr. Mohammad Nour**
Range algorithms comparison report   **Date: Sat - 19 Oct 2019**

<u>Results:</u>

      After running each algorithm and measure the time it takes to complete the task, it results the following, where N is the number of numbers in the array:

N	time (ms)		
	Algorithm 1	Algorithm 2	Algorithm 3
1000	9.236383	5.454544	0.040942
2000	8.379806	4.869934	0.078804
4000	9.939414	10.374938	0.145929
8000	35.591281	20.094953	0.289925
16000	142.429557	78.967454	0.547765
32000	573.099036	312.395491	0.801028
64000	2388.314652	1243.214705	0.370597
128000			0.712353
256000			0.25461
512000			0.303729
1024000			0.606819
2048000			1.239308
4096000			2.407794
8192000			4.864454
16384000			9.617369
32768000			18.382383
65536000			38.928968
131072000			79.661588
262144000			160.027967

Here is a chart that illustrate the growth of each algorithm:
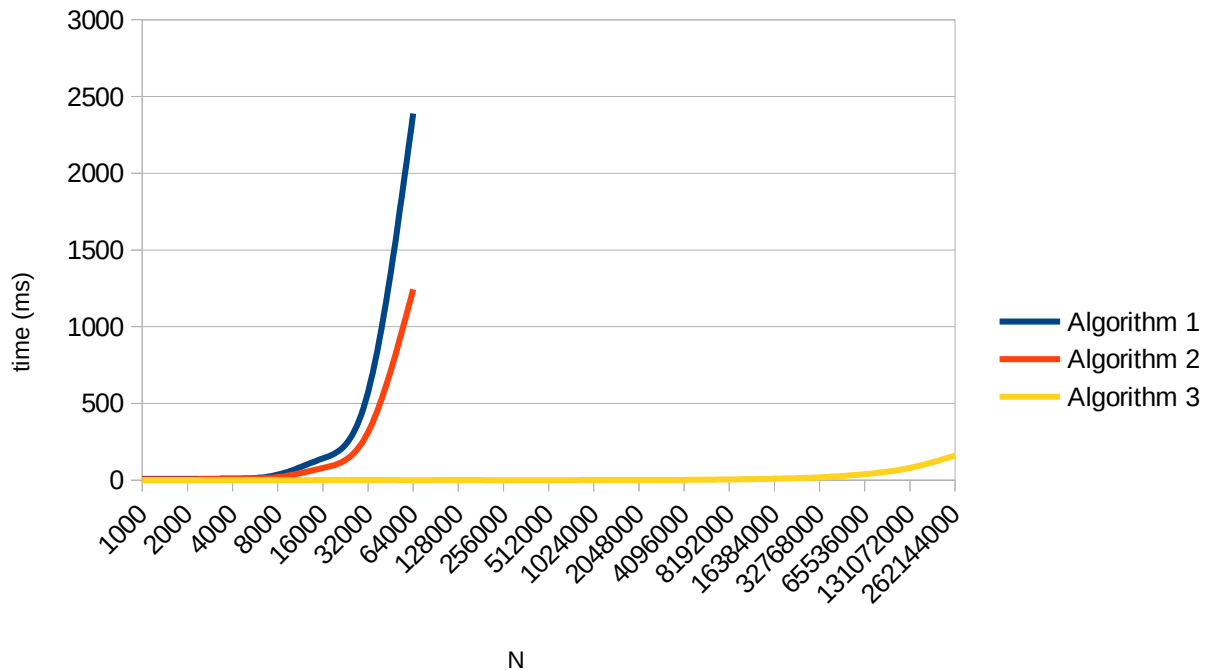
Conclusion:

We can note that the growth of 3rd algorithm is much lower than other two algorithm.

And since 3rd algorithm is order of $O(N)$, and other two algorithms are order of $O(N^2)$, we can conclude that order of $O(N)$ algorithms is better than $O(N^2)$ in terms of performance.