Analysis of ACME and AJAX Sorting Algorithms

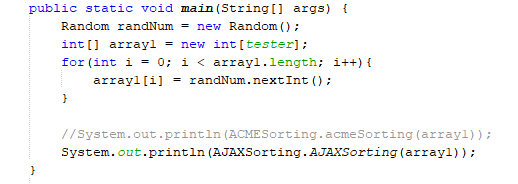
Kayla Holloway

ACME and AJAX are software development companies that have developed sorting algorithms with pricing based on the usage of their software. Columbus Software Solutions is looking to purchase either software based on their average daily records to sort. The purpose of this report is to analyze time complexity and costs in both runtime and monetary.

# **Experiment Description:**

Using the .jar files supplied, I created an array in which the size is populated with random numbers modeling average recorded sizes and information. I then recorded ordered pairs of input size to instructions and created a graph to match time complexity.

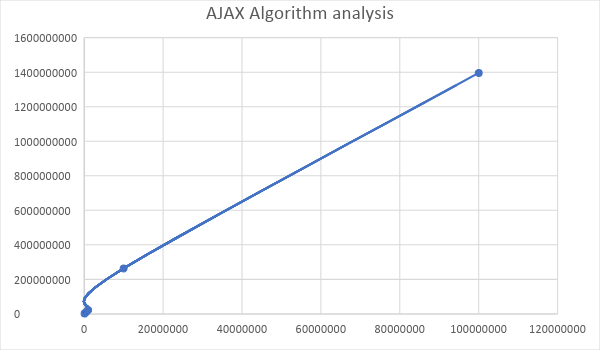
**Source code used**



# **AJAX Experiment Data**

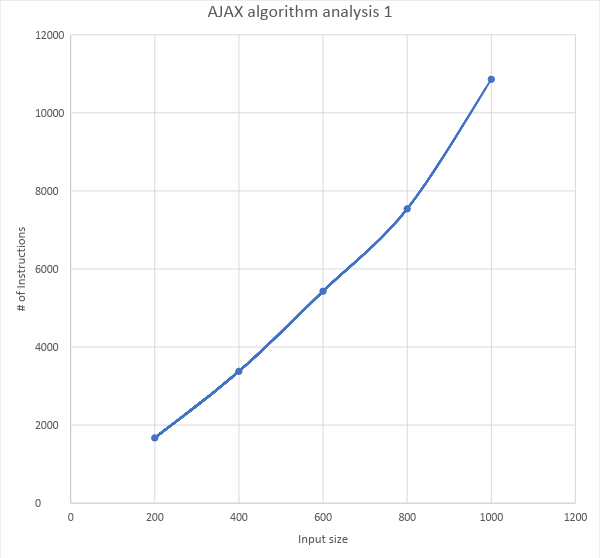
Large Input Size Test

* For an input size 100,000 it takes AJAX 1,795,882 instructions
* For an input size 500,000 it takes AJAX 10,141,285 instructions
* For an input size 1,000,000 it takes AJAX 21,317,132 instructions
* For an input size 10,000,000 it takes AJAX 263,484,126 instructions
* For an input size 100,000,000 it takes AJAX 1,395,419,630 instructions



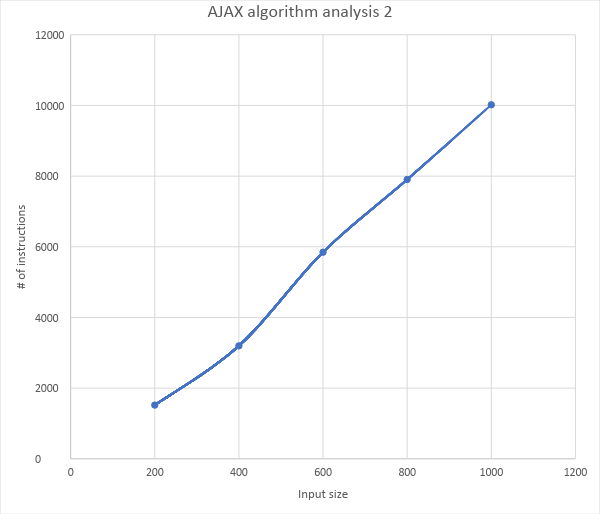
Model Input Sizes

Experiment 1:

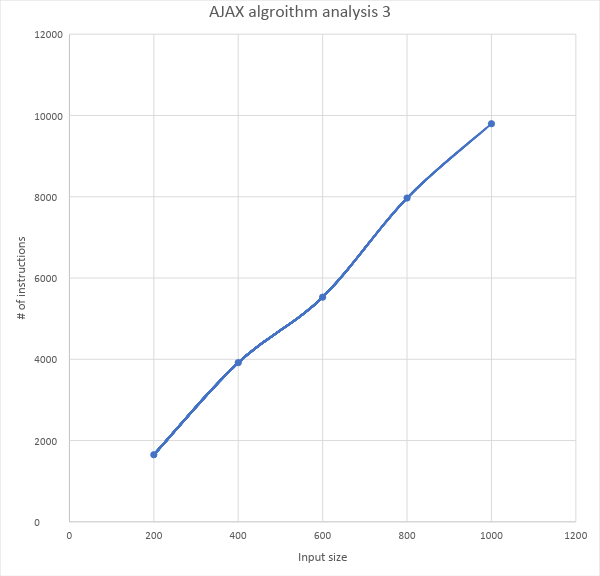
* For an input size 200; 1671 instructions.
* For an input size 400; 3377 instructions.
* For an input size 600; 5430 instructions.
* For an input size 800; 7539 instructions.
* For an input size 1000; 10859 instructions.

Experiment 2:

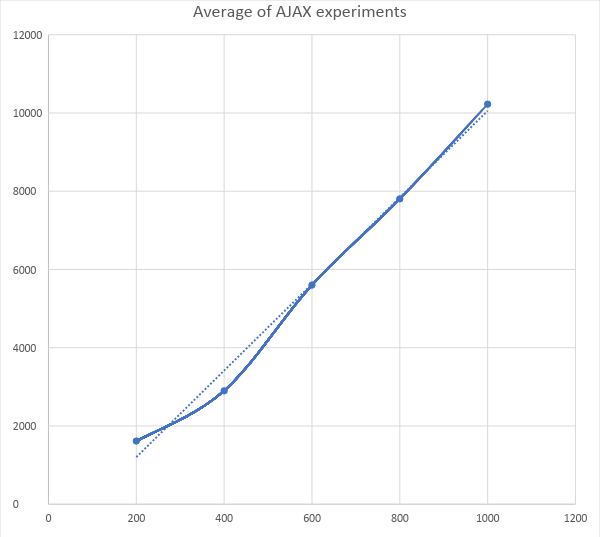
* For input size 200; 1521 instructions
* For input size 400; 3197 instructions
* For input size 600; 5847 instructions
* For input size 800; 7904 instructions
* For input size 1000; 10019 instructions



Experiment 3

* For input size 200; 1651 instructions
* For input size 400; 3917 instructions
* For input size 600; 5529 instructions
* For input size 800; 7968 instructions
* For input size 1000; 9796 instructions

**Averages:**

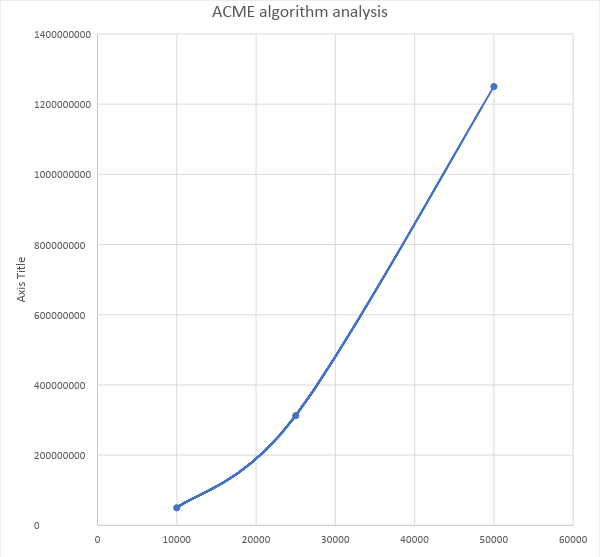
* For input size 200; 1614.3 instructions
* For input size 400; 2897 instructions
* For input size 600; 5602 instructions
* For input size 800; 7803.6 instructions
* For input size 1000; 10224.6 instructions

**Final analysis of AJAX algorithm:**

The numbers are proportionate to the increase of input size and to the number of instructions, meaning I can determine that the average time complexity for AJAX sorting algorithm is O(n) linear time complexity. The output (number of instructions) was the variable every time the experiment was ran due to random number generation. Therefore, I took the averages to produce the final graph.

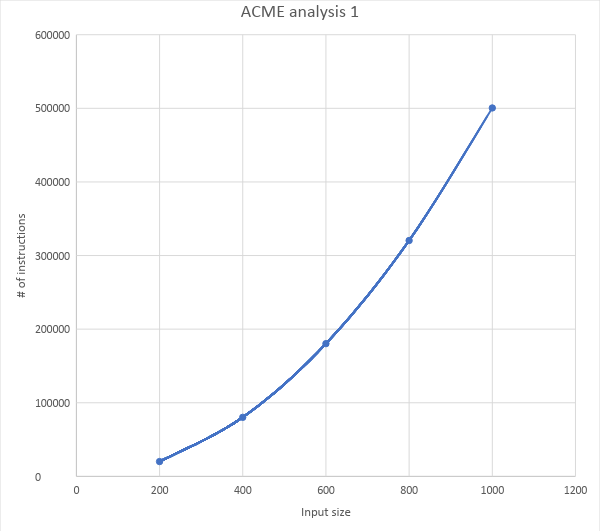
# **ACME Experiment Data**

Large input sizes:

* For an input size 10,000 it takes ACME 50,005,000 instructions
* For an input size 25,000 it takes ACME 312,512,500 instructions
* For an input size 50,000 it takes ACME 1,250,025,000 instructions

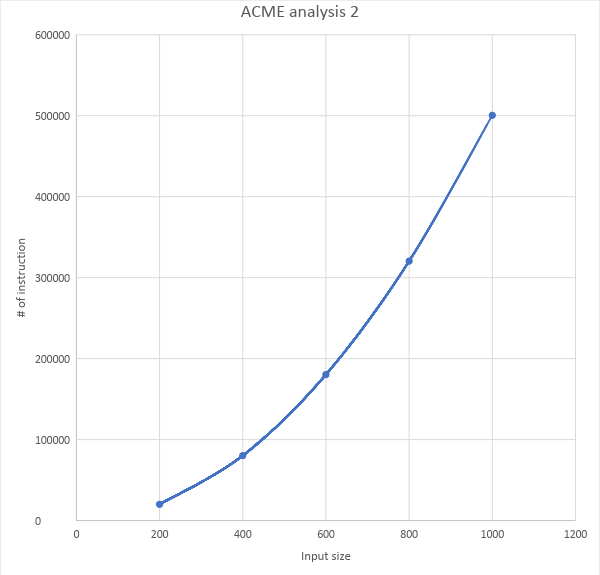
Model input sizes

Experiment 1:

* For an input size 200; 20100 instructions
* For an input size 400; 80200 instructions
* For an input size 600; 180300 instructions
* For an input size 800; 320400 instructions
* For an input size 1000; 500500 instructions

Experiment 2:

* For input size 200; 20100 instructions
* For input size 400; 80200 instructions
* For input size 600; 180300 instructions
* For input size 800; 320400 instructions
* For input size 1000; 500500 instructions



**Final analysis**

After conducting the experiment three times the data showed that the number of instructions is consistent for input size regardless of the order in the array. The rate at which the number of instructions changed between input sizes is shrinking following the trend of (n log n).

Out of all the experiments done, the output remained consistent.

|  |  |  |
| --- | --- | --- |
| Inputs | Number of instructions | Rate of increase |
| 200 | 20100 | N/A |
| 400 | 80200 | 3.9900 |
| 600 | 180300 | 2.2481 |
| 800 | 320400 | 1.777 |
| 1000 | 500500 | 1.5621 |
| 1200 | 720600 | 1.4397 |
| 1400 | 980700 | 1.3609 |
| 1600 | 1280800 | 1.306 |

…

|  |  |  |
| --- | --- | --- |
| 25000 | 312512500 | 1.008048 |

# **Comparing Both Algorithms**

General Cost of Run time (over the interval 200-1000 incrementing by 200):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algorithm | Input size: 200 | 400 | 600 | 800 | 1000 |
| AJAX | 1614.3 | 2897 | 5602 | 7803.6 | 10,244.3 |
| ACME | 20,100 | 80,200 | 180,300 | 320,400 | 500,500 |

Cost of run time based on daily averages ( number of instructions):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday |
|  | 10 | 1 | 30 | 50 | 0.5 | 100 | 0.1 |
| AJAX | 263,484,126 | 21,317,132 | 817,402,203 | 1,415,014,439 | 10,141,285 | 1,395,419,630 | 1,795,882 |
| ACME: | 7.05E+0 | 7.05E+9 | 2.14E+11 | 3.53E+11 | 1.24E+10 | 7.05E+11 | 5.0005E+8 |

Cost in USD ($)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday |
| AJAX | $10,000,000 | $1,000,000 | $30,000,000 | $50,000,000 | $500,000 | $100,000,000 | $100,000 |
| ACME | $130,000 | $13,000 | $390,000 | $650,000 | $13,000 | $1,300,000 | $13,000 |

# **Conclusion**

The variance in the two algorithms is quality versus cost. AJAX costs more, however it is more efficient and saves time, while ACME costs less but takes extra time to process requests. Now that all the data has been presented, it is up the managers of Columbus Software Systems to decide on which software to implement.