

# Arraylist Functionality

Eric Chrisman

# What is the Project About?

- Arraylist is a tool for easier array management
- It can be used to add and remove elements from the array
- This project tests the speed of Arraylist

40	55	63	17	22	68	89	97	89
0	1	2	3	4	5	6	7	8

<- Array Indices

**Array Length = 9**

**First Index = 0**

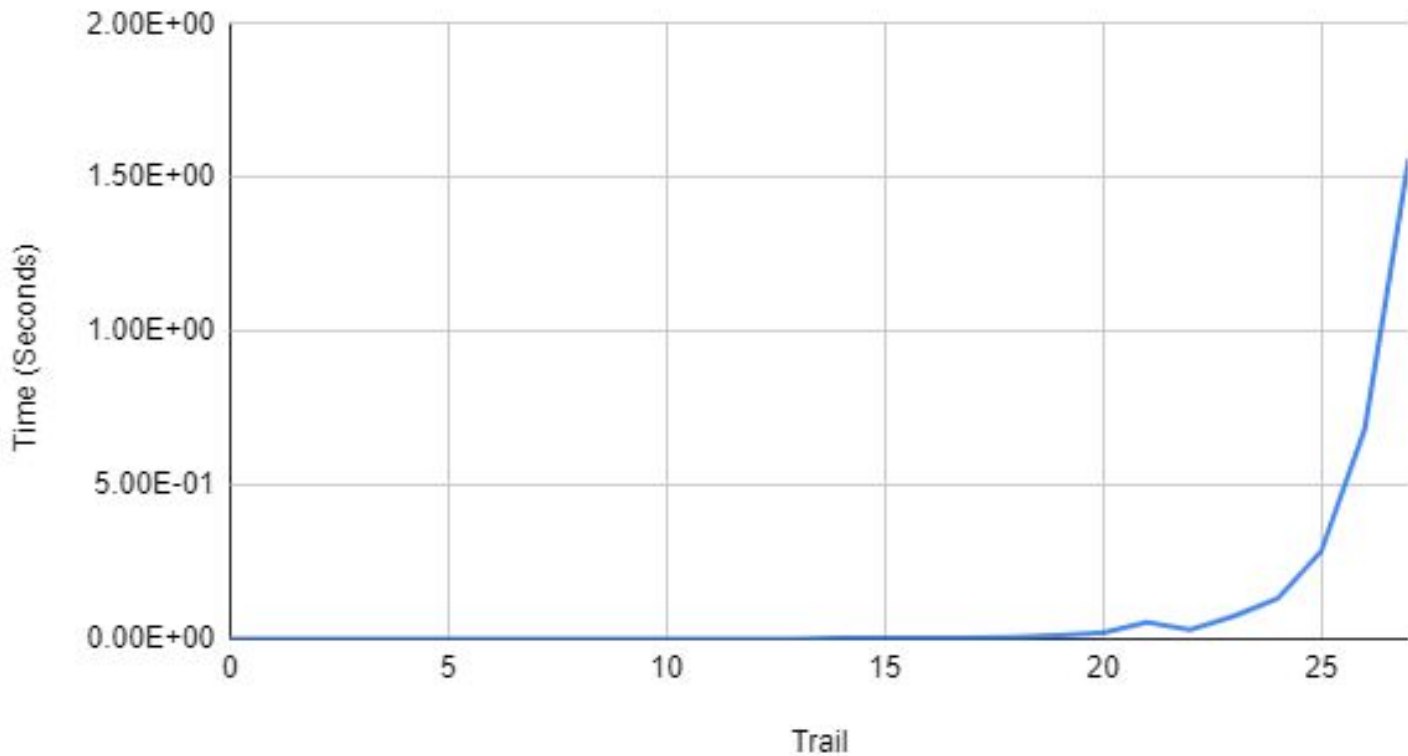
**Last Index = 8**

# Speed of Adding Numbers

- Each trail has  $2^x$  amount of numbers
- $x$ 's value starts at zero and goes up by one every trail
- The numbers are added to the array one at a time
- The time it takes to add up all the numbers is recorded
- Trails ends either after ten minutes pass or the computer runs out of memory

Trial ended from low computer memory.

### Processing Time for Adding onto the Array



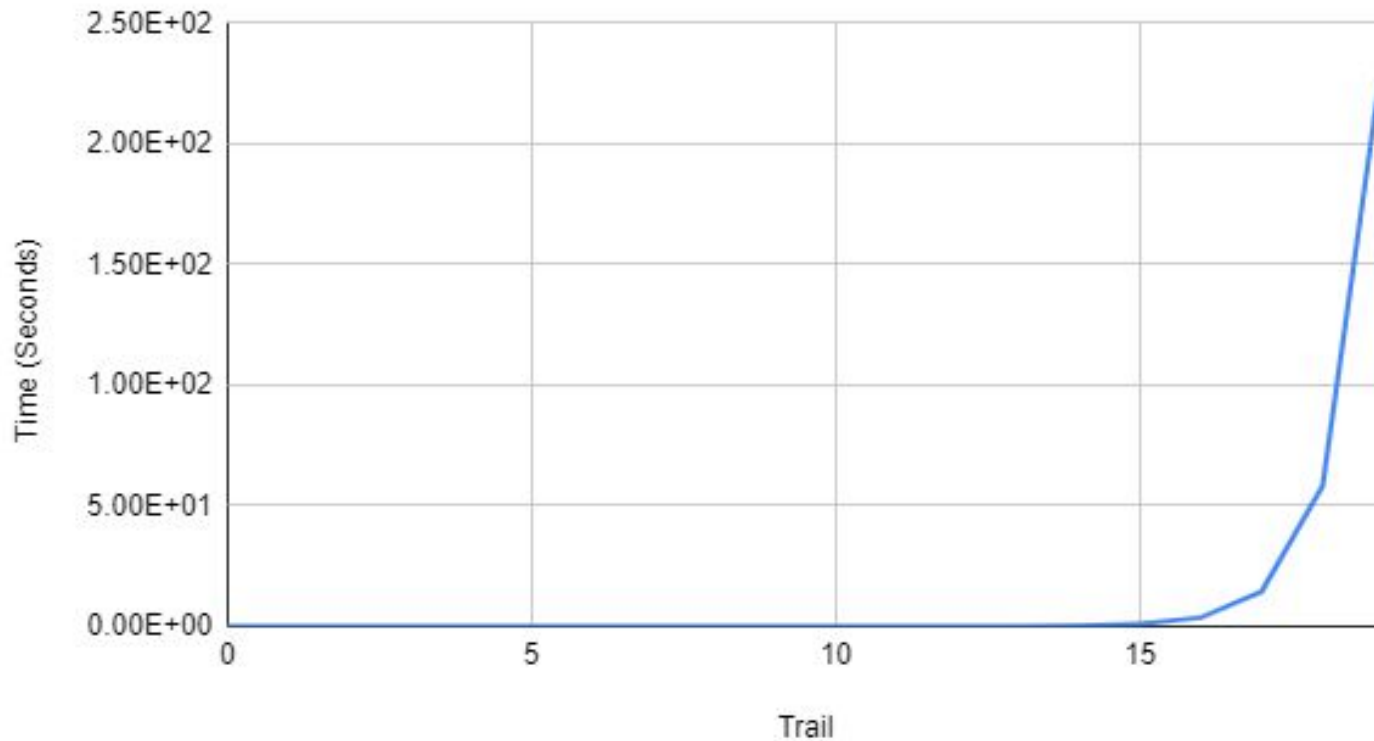
Program lasted for 27 trails. Last trail took 1.56 seconds to complete.

# Speed of Deleting Numbers

- This test mimics the same steps as the adding test
- When all the numbers are added a timer starts
- The program will remove the first element; one at a time
- Trails end either when ten minutes pass or the computer runs out of memory

Trail ended after ten minutes.

### Processing Time for Removing Numbers



Program lasted for 19 trails. Last trail took 237.39 seconds to complete.

# Analysis

- Adding was much faster than deleting terms
- All adding trials took less than two seconds
- The deleting trials lasted till the ten minute mark
- The adding trials ended from memory shortage
- The adding trials had eight more trials than the deleting ones

# Conclusion

- The deleting trails took longer because removing the first term was not effective
- When the first is deleted, the size of the array is decreased
- When this happens, all terms are moved back one index value
- This caused the major time difference between the two test
- To be more effective, the last term should be removed instead of the first term.