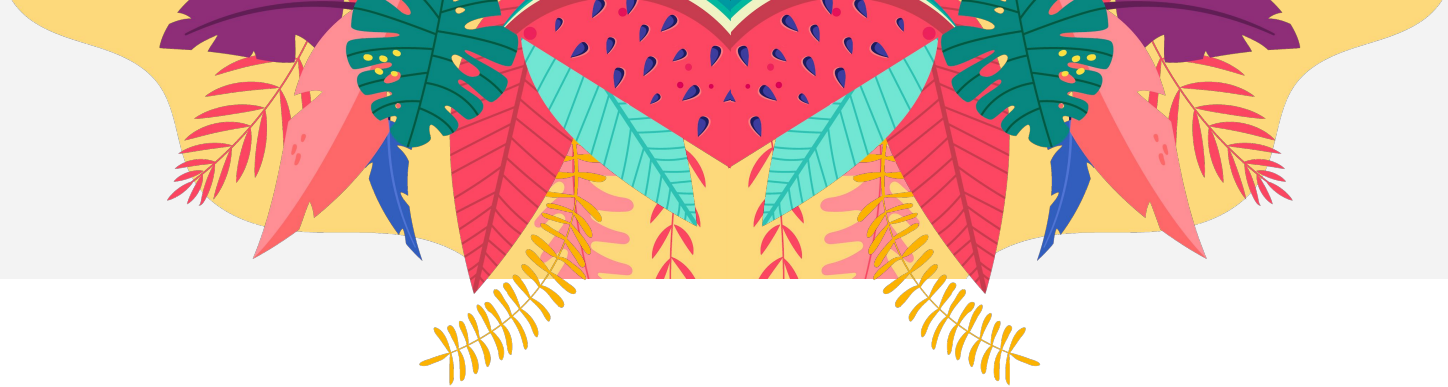
Isha Patel



DESCRIPTION

This presentation compares the amount of time it takes to add 2^i unique items for all non-negative integer values of i to a list and the amount of time it takes to remove 2^i unique items from the front of the ArrayList for all non-negative integer values of i.

Both of these calculations were made until the until the computer ran out of memory or the time exceeded ten minutes.

Figure 1

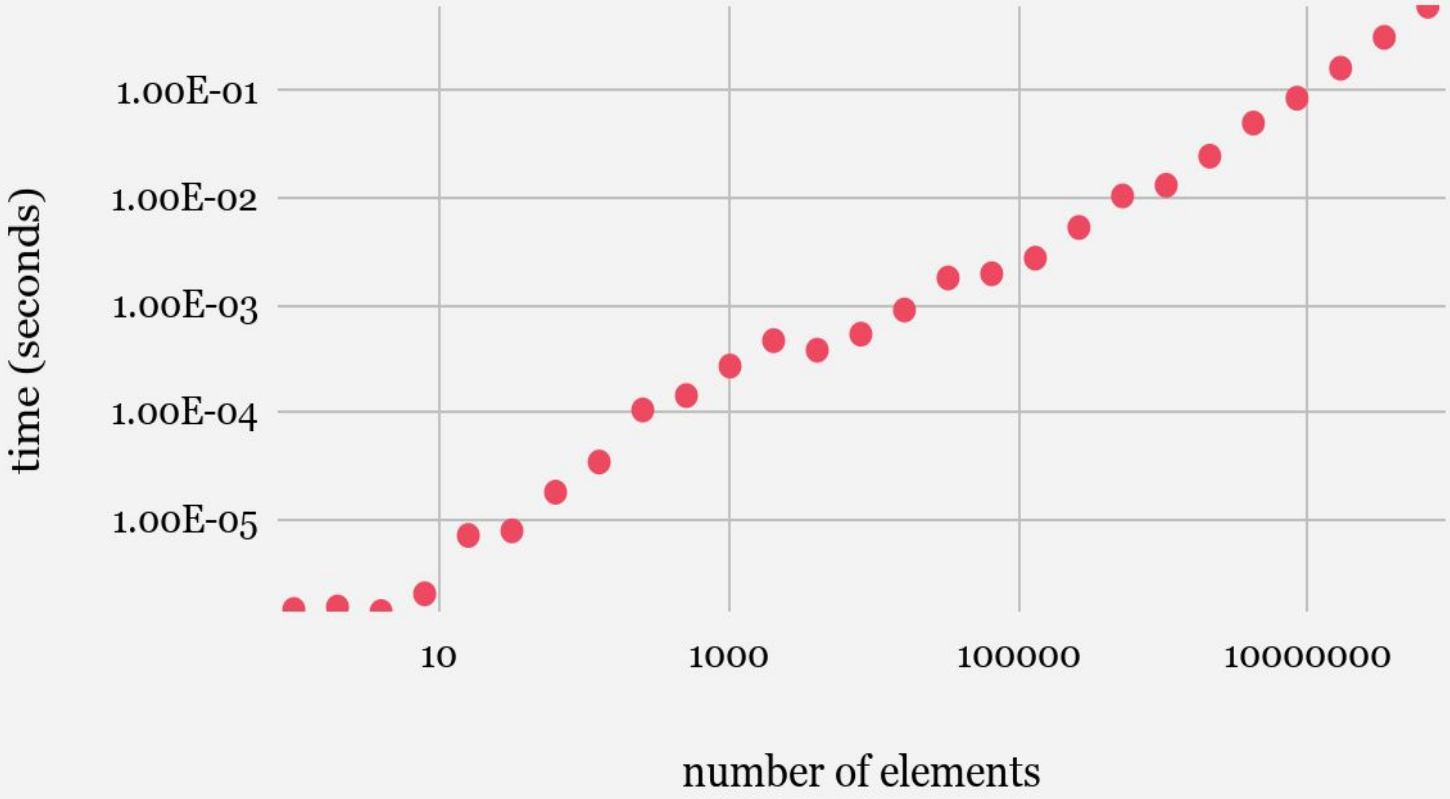

Time It Takes to Add 2^i Elements
time (seconds)
1.00E-01
1.00E-02
1.00E-03
1.00E-04
1.00E-05
10
1000
100000
10000000
number of elements

Figure 2

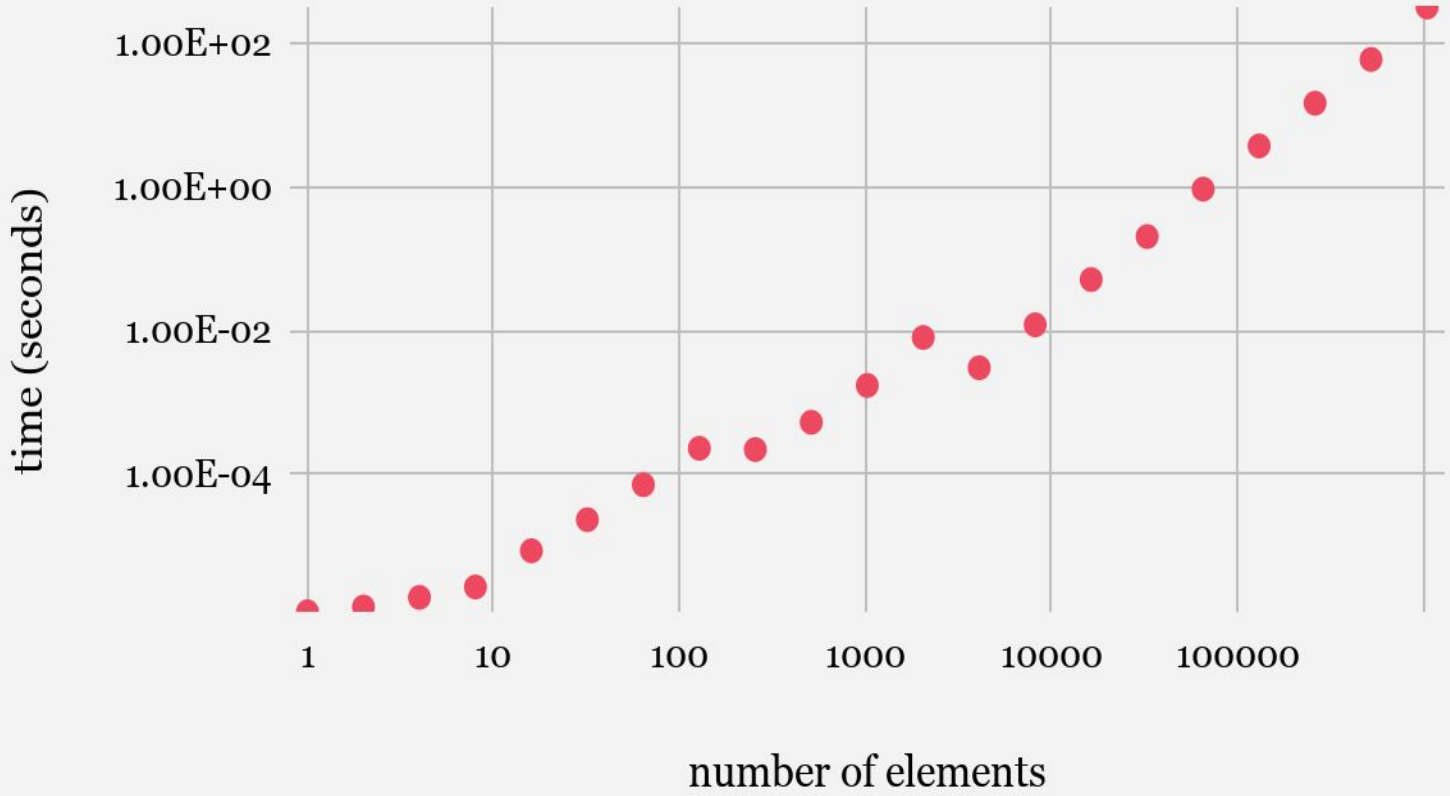
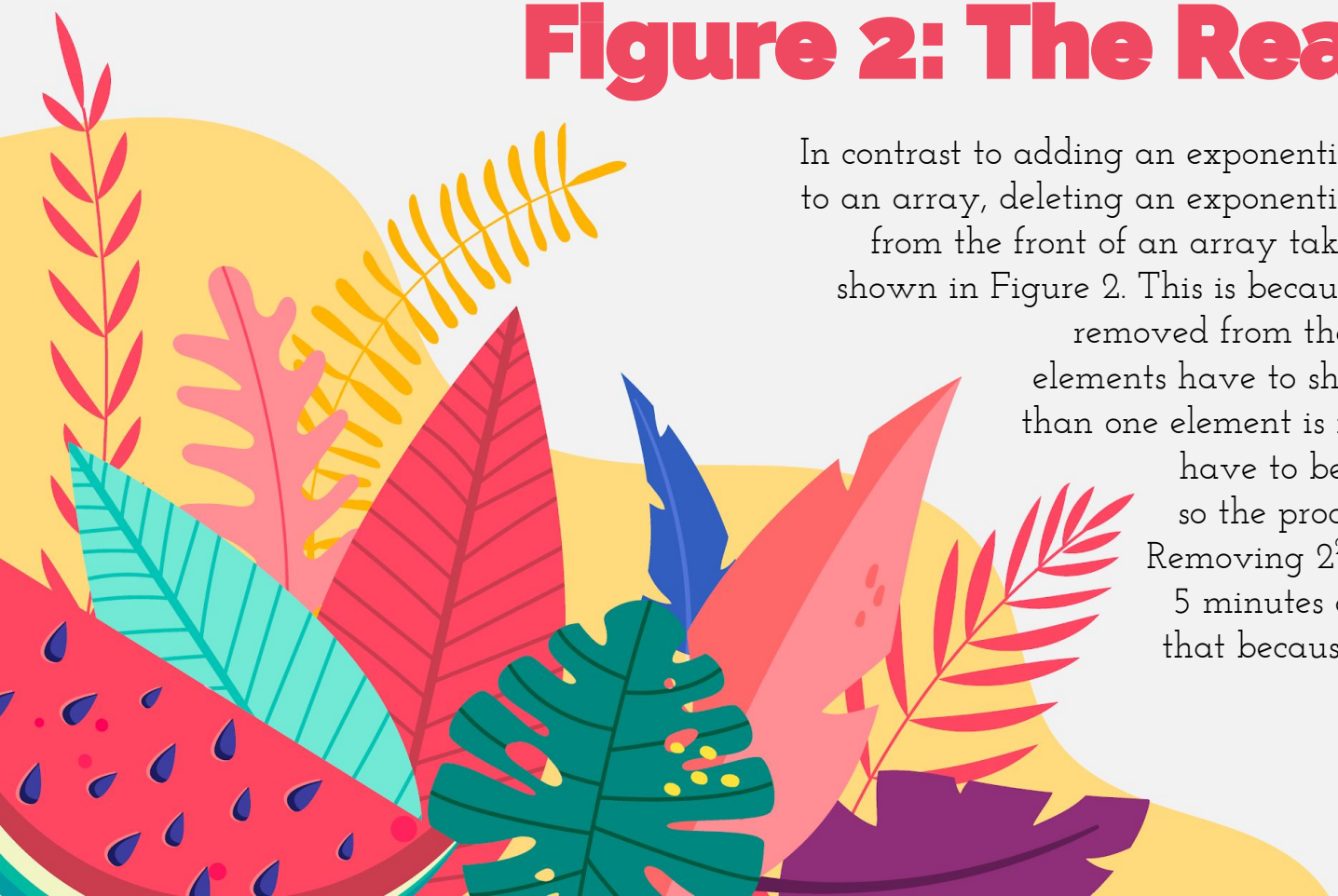

Time It Takes to Delete 2^i Elements
1.00E+02
1.00E+00
1.00E-02
1.00E-04
time (seconds)
1
10
100
1000
10000
100000
number of elements

Figure 1: The Reasoning

As more and more elements were being added to the list, it took longer to make the array as shown in Figure 1. I was able to generate an array with 2^{27} elements before I ran out of memory. However, all of those arrays were generated under one second, so if I had more memory then I would have surely been able to generate many more arrays that had exponentially higher number of elements.

Figure 2: The Reasoning



In contrast to adding an exponential number of elements to an array, deleting an exponential number of elements from the front of an array takes much more time, as shown in Figure 2. This is because when an element is removed from the front, all of the other elements have to shift up and when more than one element is removed, the elements have to be removed one by one, so the process takes much longer. Removing 2^{20} elements took about 5 minutes and I had to stop after that because of the long length of time.


THANKS

CREDITS

- Presentation template by Slidesgo
- Icons by Flaticon
- Infographics by Freepik
- Images created by Freepik - Freepik
- Author introduction slide photo created by Freepik
- Text & Image slide photo created by Freepik.com