

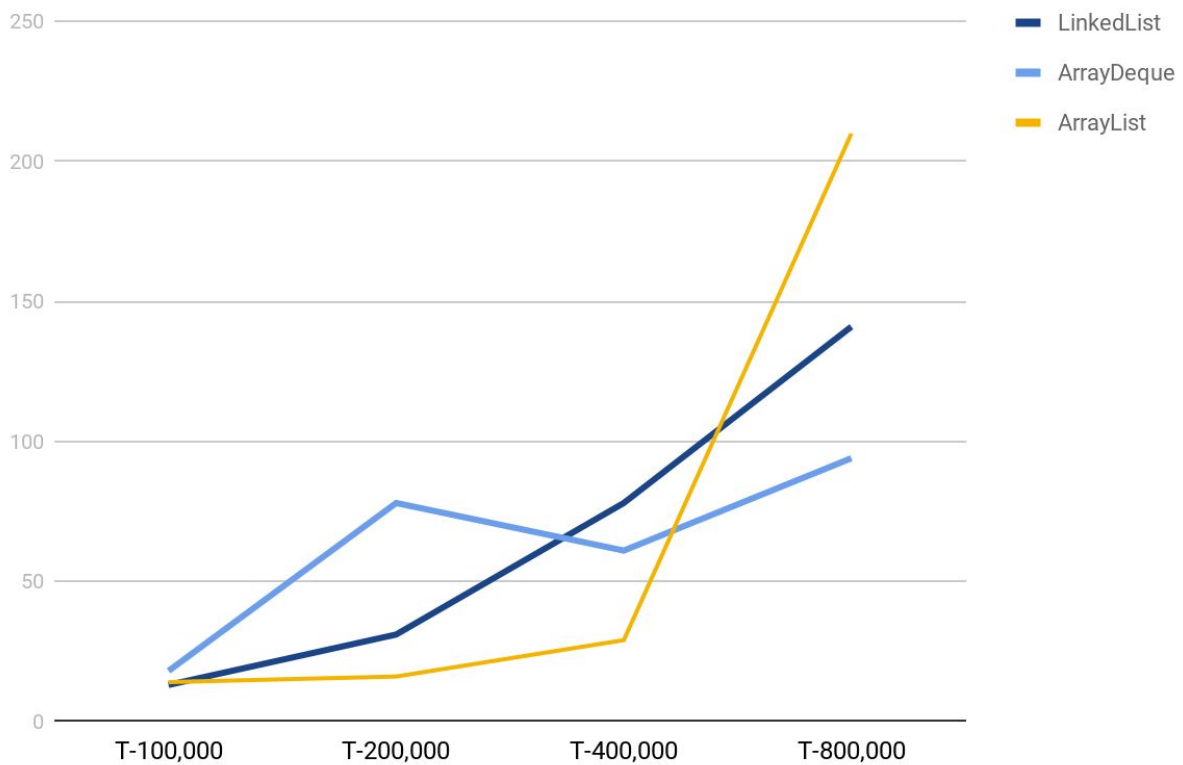
Traae Bloxham
Dr. Kerby
Data Structures and Algorithms
3/13/2020

ArrayDeque VS ArrayList VS LinkedList

All results are time in milliseconds

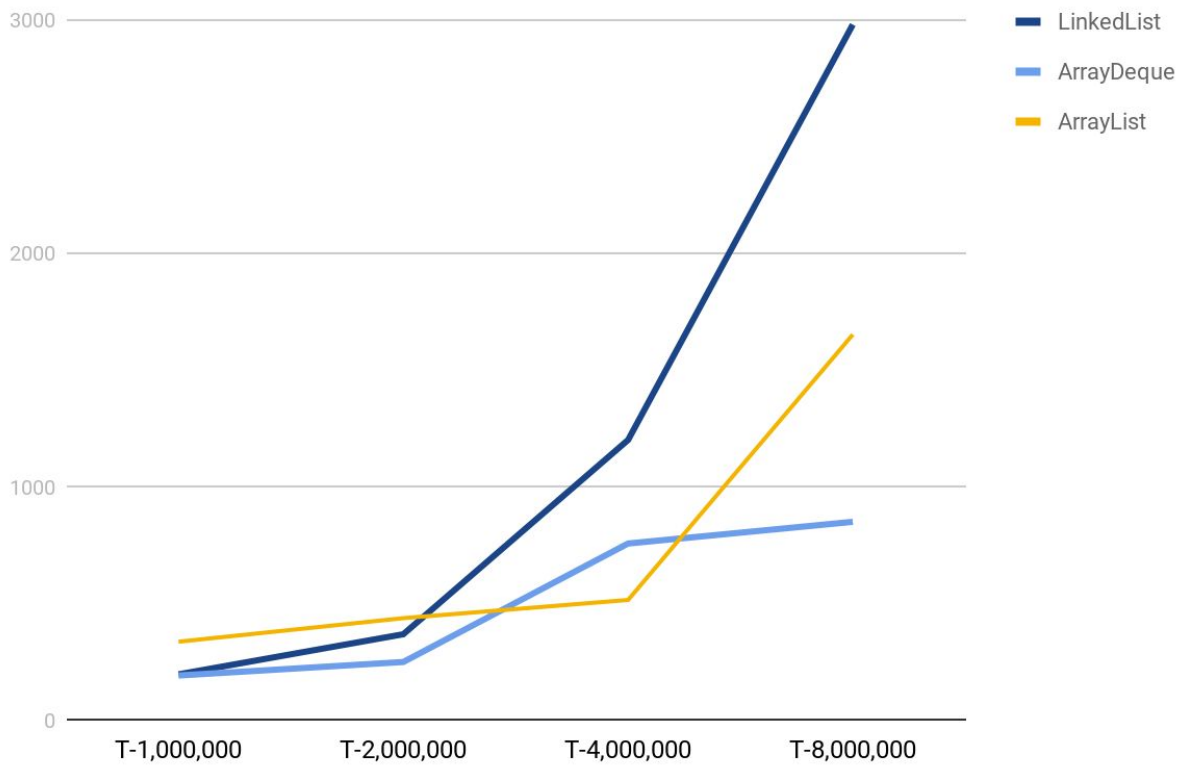
Adding to front:

| | 100,000 | 200,000 | 400,000 | 800,000 |
|------------|---------|---------|---------|---------|
| LinkedList | 13 | 31 | 78 | 141 |
| ArrayDeque | 18 | 78 | 61 | 94 |
| ArrayList | 14 | 16 | 29 | 210 |



Adding to back:

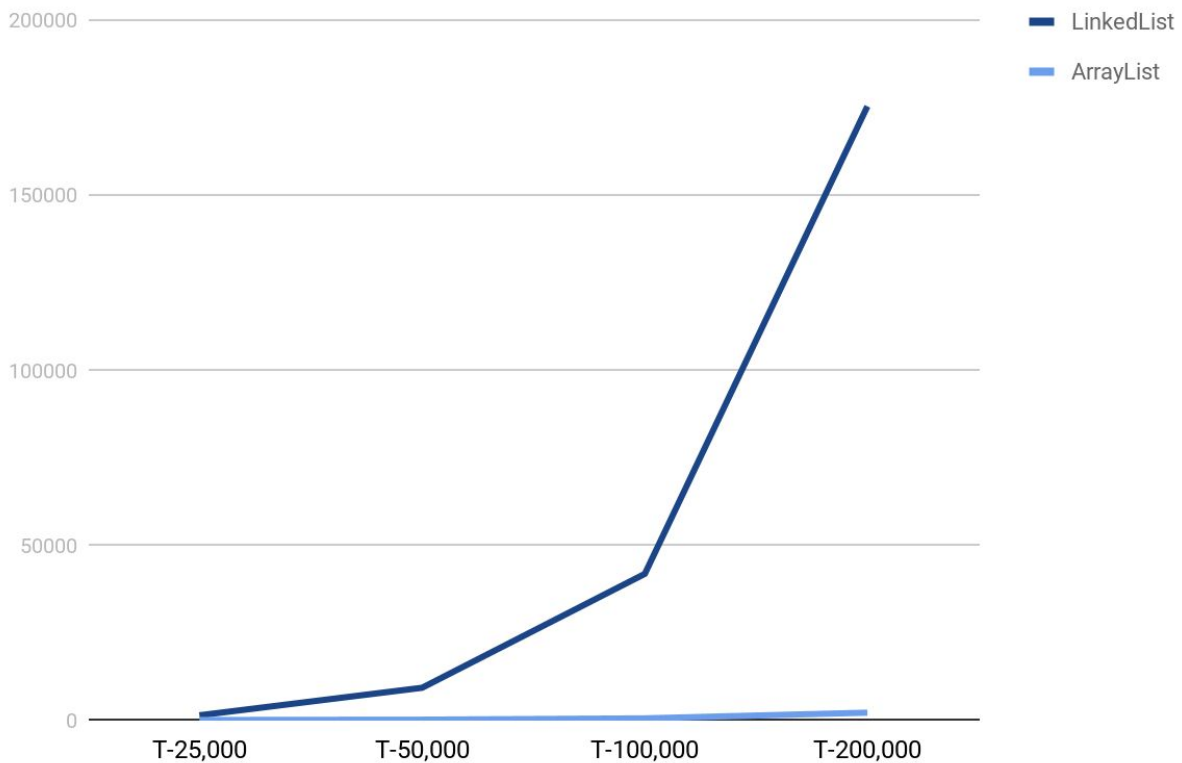
| | 1,000,000 | 2,000,000 | 4,000,000 | 8,000,000 |
|------------|-----------|-----------|-----------|-----------|
| LinkedList | 197 | 369 | 1202 | 2983 |
| ArrayDeque | 192 | 250 | 758 | 851 |
| ArrayList | 337 | 438 | 516 | 1654 |



Adding to middle:

| | 25,000 | 50,000 | 100,000 | 200,000 |
|-------------|--------|--------|---------|---------|
| LinkedList | 1428 | 9296 | 41850 | 175509 |
| ArrayDeque* | X | X | X | X |
| ArrayList | 83 | 176 | 548 | 2179 |

*arrayDeque doesn't add to the middle, excluded from the test.



Analysis:

At the highest trial volumes, the ArrayDeque is supreme at adding to the front and back. Meanwhile, the ArrayList is far and away better at adding to the middle, as compared to the LinkedList. It seems that using the ArrayDeque would be optimal for anything related to a queue or stack operation, and the ArrayList for anything where modifying the inside is necessary.

The LinkedList didn't come in 1st place on any of the operations when working at a high volume, but it was significantly faster than ArrayDeque on the 100,000 and 200,000 trials. It seems the LinkedList will be faster at a lower volume than the ArrayDeque.