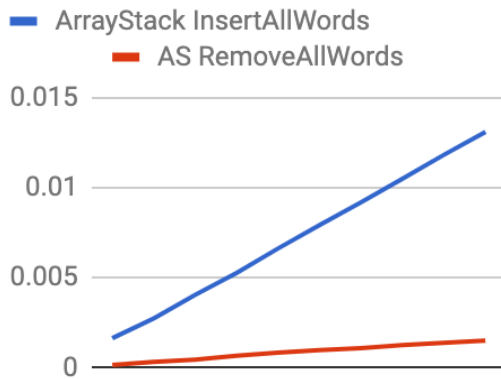


ArrayStack:

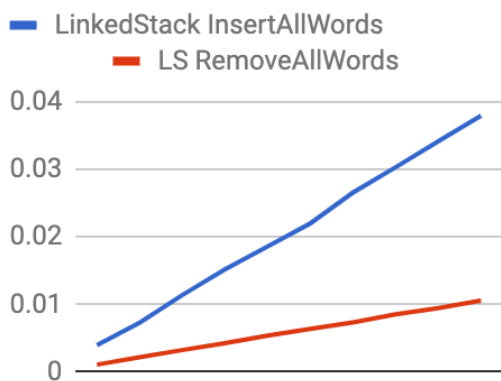


Observed time complexity:

Insert: $O(1)$. Inserting n words takes $O(1) * n = O(n)$ time, as observed in the graph.

Remove: $O(1)$. Removing n words takes $O(1) * n = O(n)$ time, as observed in the graph.

LinkedListStack:

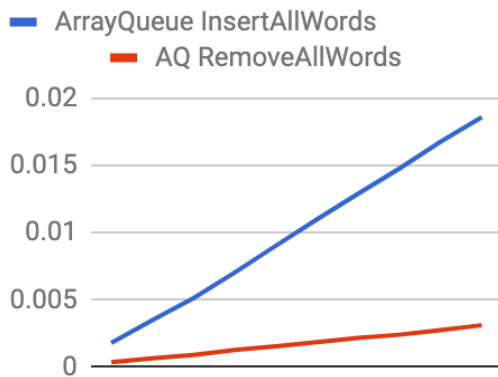


Observed time complexity:

Insert: $O(1)$. Inserting n words takes $O(1) * n = O(n)$ time, as observed in the graph.

Remove: $O(1)$. Removing n words takes $O(1) * n = O(n)$ time, as observed in the graph.

ArrayQueue:

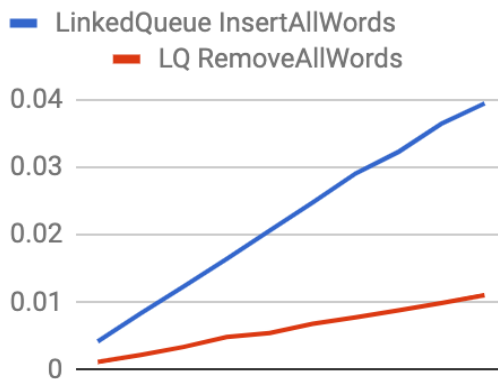


Observed time complexity:

Insert: $O(1)$. Inserting n words takes $O(1) * n = O(n)$ time, as observed in the graph.

Remove: $O(1)$. Removing n words takes $O(1) * n = O(n)$ time, as observed in the graph.

LinkedListQueue:



Observed time complexity:

Insert: $O(1)$. Inserting n words takes $O(1) * n = O(n)$ time, as observed in the graph.

Remove: $O(1)$. Removing n words takes $O(1) * n = O(n)$ time, as observed in the graph.