Data Structures & Algorithms

Sep 19, 2018

Exo: 1.4.43 (Page 214)

Run experiments to validate the hypothesis that resizing arrays are faster than linked lists for stacks.

Authors: Cecilia Sui, Derek Dyer, Maeve Mueller

Stacks backed by a singly-linked list:

A linked list supports O(1) time prepend and delete-first prepend, and therefore the cost to push or pop elements into a linked-list stack is also O(1) (in the worst-case). Each new item added to the list requires a new allocation, and allocations (compared to other operations) can be expensive.

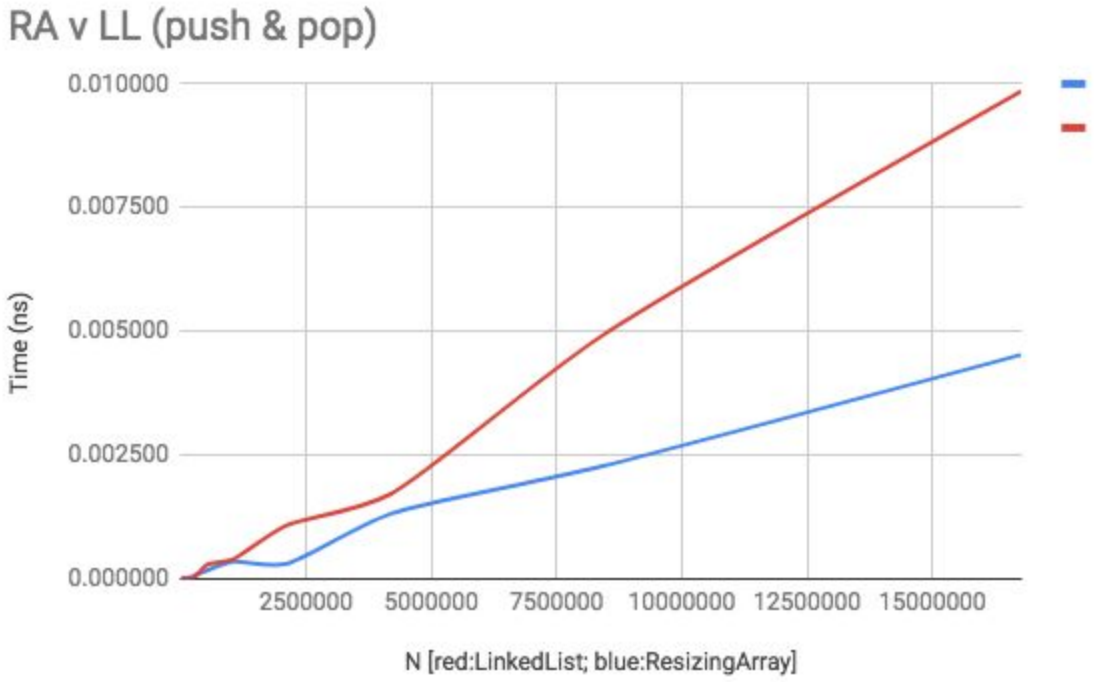
Stacks backed by resizing arrays:

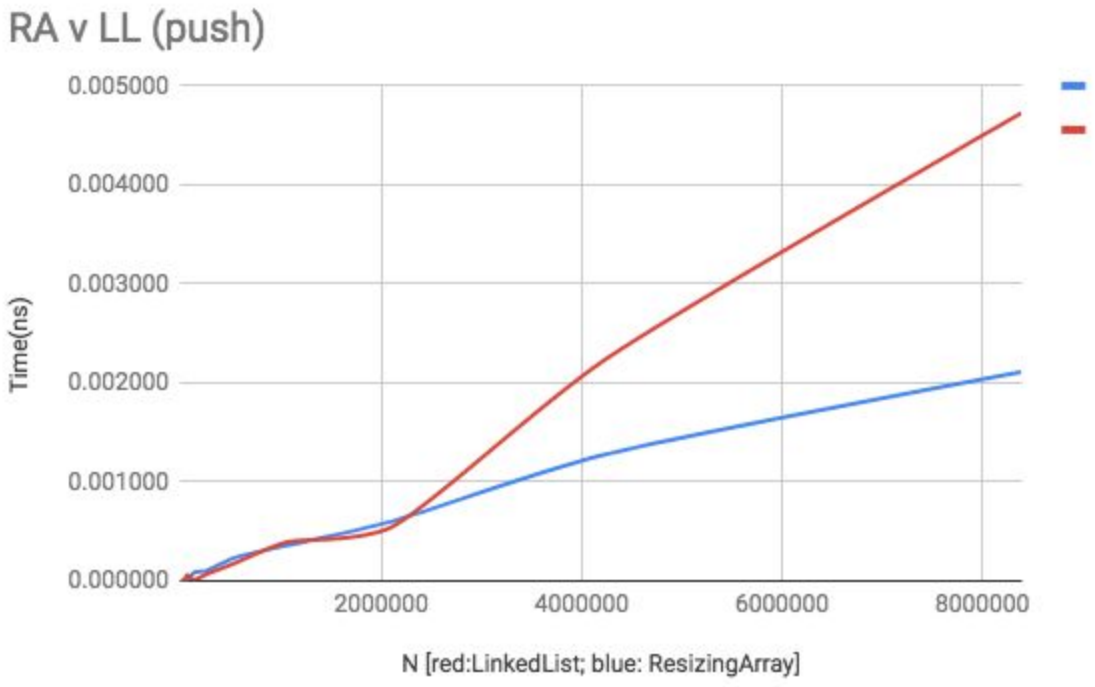
Pushing elements onto the stack can be executed by appending an element to the array. This calculates the sum O(1) time and the worst-case O(n) time. In order to remove the last element in the array, the pop function can be used. This runs in the worst-case O(1) (or sum of O(1) if wanting to hold unoccupied space).

Conclusion:

In our data it is clear to see that in each function, resizing an array is faster than a linked list. While push and pop are executed, the times of each the array and linked list are initially similar but, the more items appended, the slower a linked list is executed. The same results are evident

when pushing items. In both cases, one can observe that the speed is able to stay on track for only so long before the linked list starts require more time to process.





Please see the chart below for the data:

N	RA-pushpop Time(ns)	LL-pushpop Time(ns)	RA - push Time(ns)	LL - push Time(ns)
1	0.000000	0.000001	0.000001	0.000001
2	0.000000	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000
8	0.000000	0.000000	0.000000	0.000000
16	0.000000	0.000000	0.000000	0.000000
32	0.000000	0.000000	0.000000	0.000000
64	0.000000	0.000000	0.000000	0.000000
128	0.000001	0.000001	0.000000	0.000000
256	0.000001	0.000000	0.000000	0.000001
512	0.000000	0.000000	0.000001	0.000000
1024	0.000000	0.000000	0.000001	0.000000
2048	0.000002	0.000001	0.000002	0.000004
4096	0.000002	0.000002	0.000001	0.000001
8192	0.000003	0.000003	0.000004	0.000002
16384	0.000004	0.000004	0.000004	0.000003
32768	0.000005	0.000005	0.000005	0.000005
65536	0.000024	0.000006	0.000011	0.000059
131072	0.000009	0.000026	0.000083	0.000011
262144	0.000066	0.000023	0.000103	0.000069
524288	0.000172	0.000282	0.000229	0.000172
1048576	0.000347	0.000397	0.000357	0.000389
2097152	0.000299	0.001073	0.000598	0.000539
4194304	0.001316	0.001715	0.001265	0.002206
8388608	0.002270	0.004896	0.002110	0.004724
16777216	0.004528	0.009848		