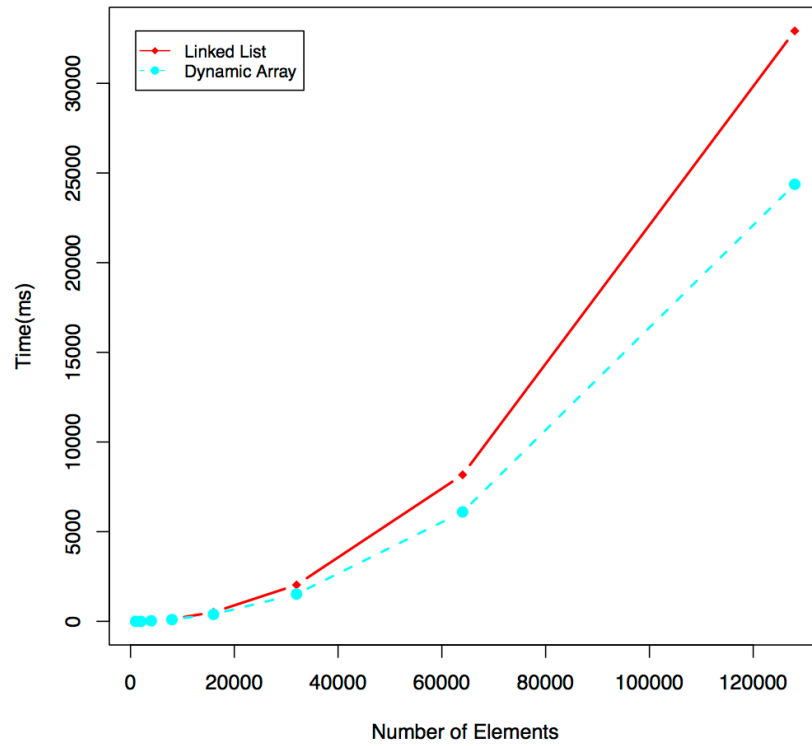
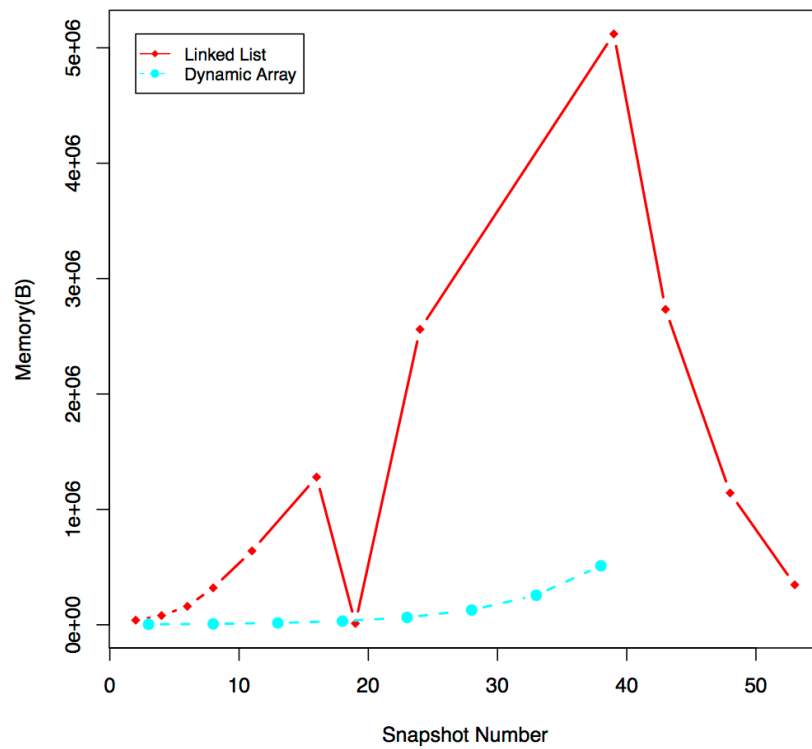


Execution Time Performance Comparison



Memory Requirement Comparison



1. Which of the implementations uses more memory? Explain why.
Linked list used more memory than dynamic array. The reason is that linked list needs extra head and tail sentinels, and two extra pointers pointed to previous link and next link for each node. While dynamic array only need to allocate enough memory for data.
2. Which of the implementations is the fastest? Explain why.
For performing n calls to `add()` and n calls to `contains()`, dynamic array is the fastest. For `add()`, dynamic array has $O(1+)$ complexity while linked list has $O(1)$ complexity. But for `contains()`, we have random access to the data in dynamic array while we have to access elements sequentially starting from the first node in linked list.
3. Would you expect anything to change if the loop performed `remove()` instead of `contains()`? If so, what?
I would expect that for linked list, `remove()` would be way more efficient than dynamic array in term of execution time since dynamic array need to copy each element after the removed element forward. But for linked list, we just need to adjust the pointers and free the removed node. In term of memory requirement, linked list still requires more memory than dynamic array.