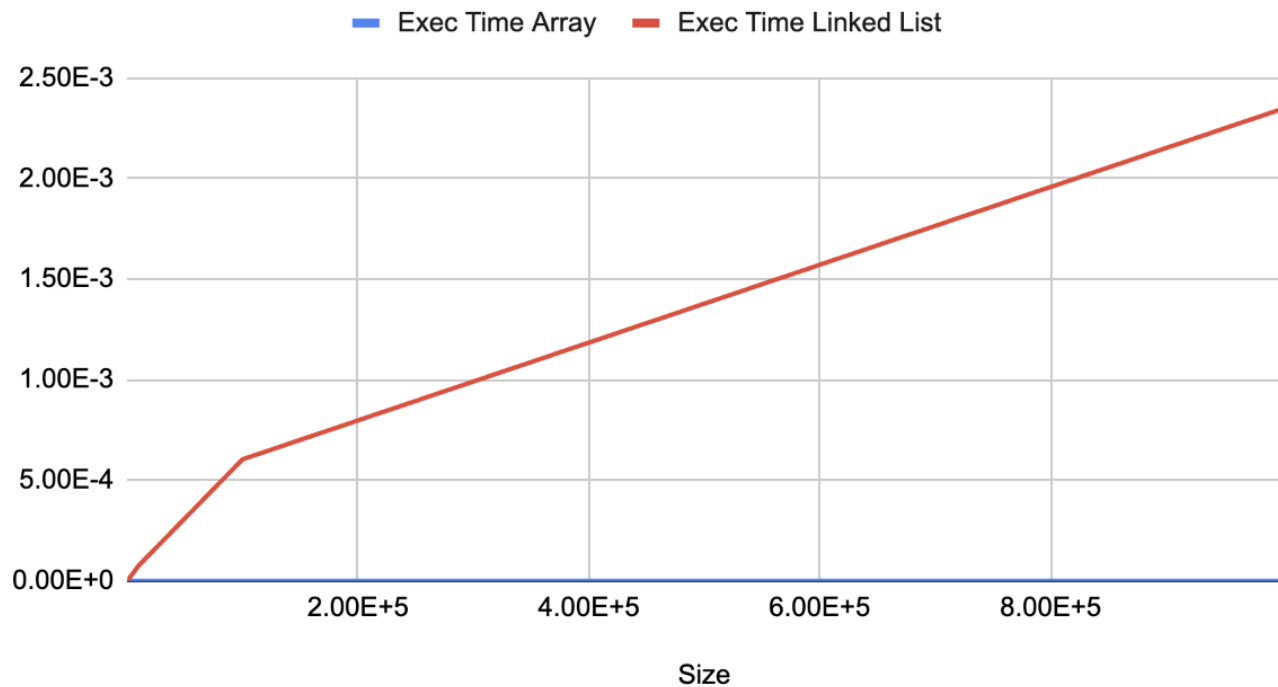


First name	Anna
Last name	Hauk
Collaborator	n/a
Pledge	I pledge my honor that I have abided by the Stevens Honor system

1 Task 1: Profiling a Linked List and an Array

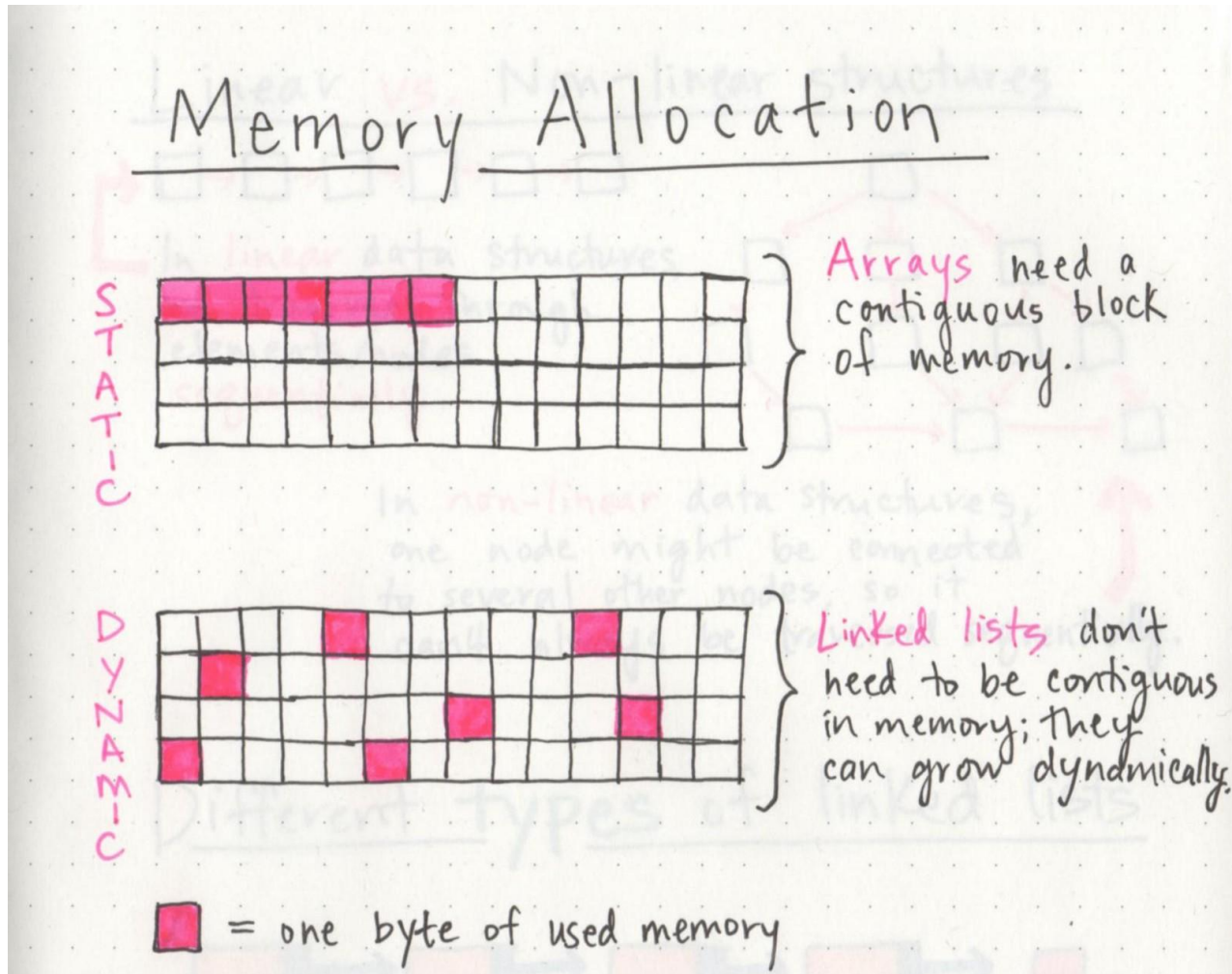
Please present your experiment record below: either a graph or a chart.

Exec Time Array and Exec Time Linked List



Please explain: why do the two algorithms with both $O(n)$ complexity, have very different performance when n increases? You need to explain in detail from the perspective of **locality**.

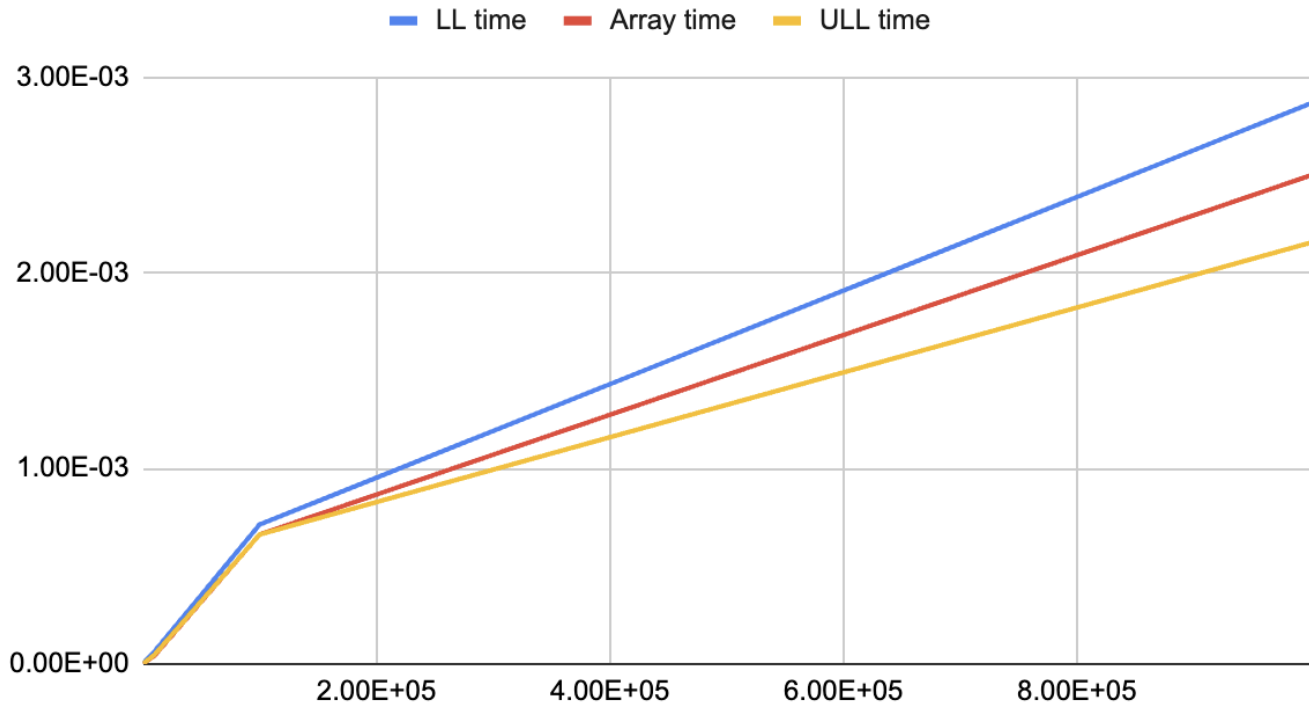
Array traversals perform better than Linked List traversals despite the same algorithmic complexity because of spatial locality. An array is a data structure where random access is possible and the elements are stored sequentially whereas linked lists aren't and have to be traversed sequentially. The image below depicts this. So arrays have better spatial locality because the memory is stored in contiguous blocks. Linked lists do not have good spatial locality because elements are scattered throughout memory.



2 Task 2: Locality Improved Linked List

Present your experiment record below: either a graph or a chart.

LL time, Array time and ULL time



Please explain: what is the time complexity of unrolled linked list? How does a unrolled linked list improve the efficiency of traversal in terms of locality?

The time complexity of an unrolled linked list traversal is still $O(n)$. The idea is that you're putting the data into arrays so the memory is consecutive and then putting those arrays into nodes for the linked list to access. There is a higher probability that you'll access nearby elements in the same node. In addition to this, there are less pointers because each node contains multiple elements. These will both improve spatial locality and cache performance because less memory locations need to be accessed to traverse the same number of elements.