CS 382
Fall 2023

Computer Architecture
and Organization

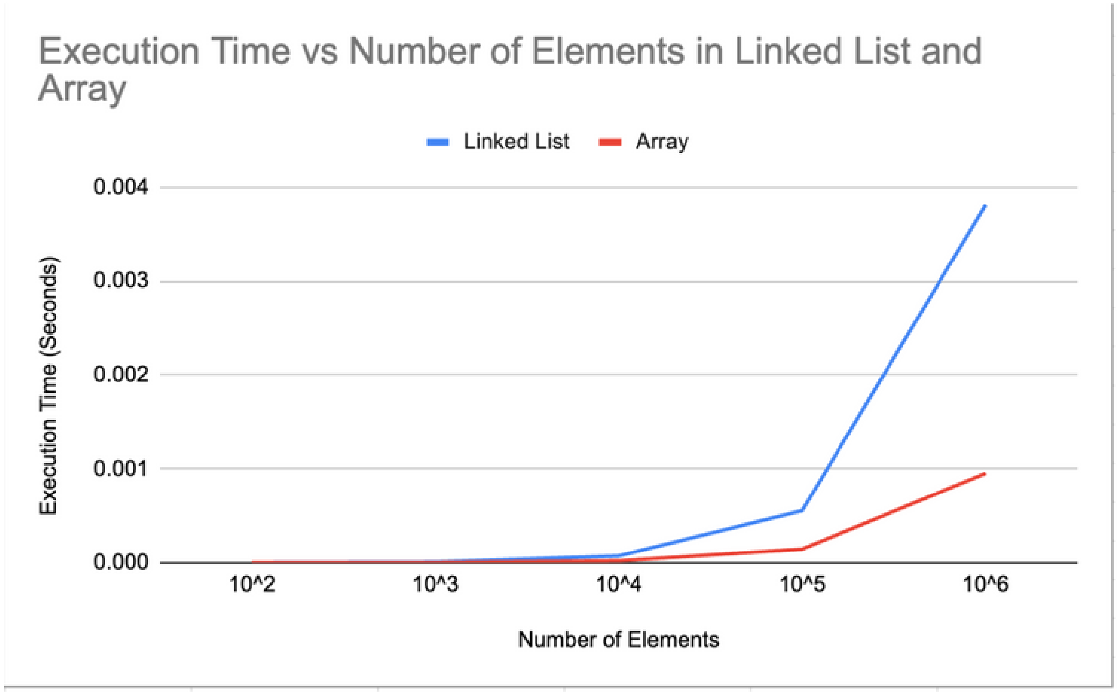
Lab 9 Report

First name	Jack
Last name	Patterson
Collaborator	Dylan Faulhaber
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.

Execution Time (Array)	Execution Time (Linked List)	Elements
0.000001	0.000002	10^2
0.000002	0.000008	10^3
0.000021	0.000074	10^4
0.000146	0.000556	10^5
0.000953	0.003811	10^6

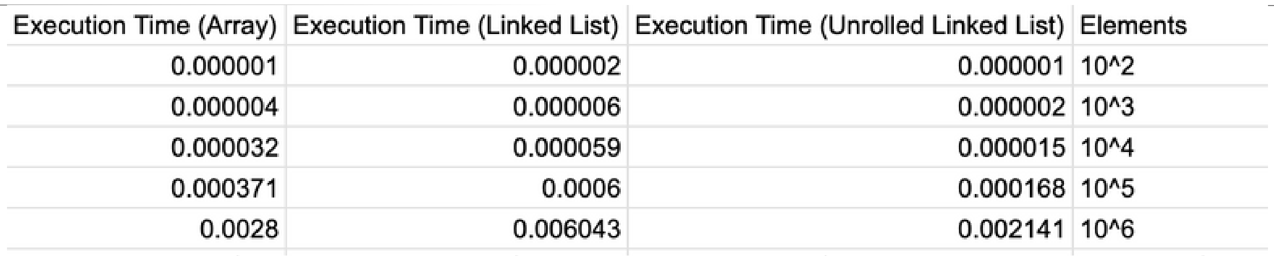


Please explain: why does the two algorithms with both $\mathcal{O}(n)$ complexity, have very different performance when n increases? You need to explain in detail from the perspective of **locality**.

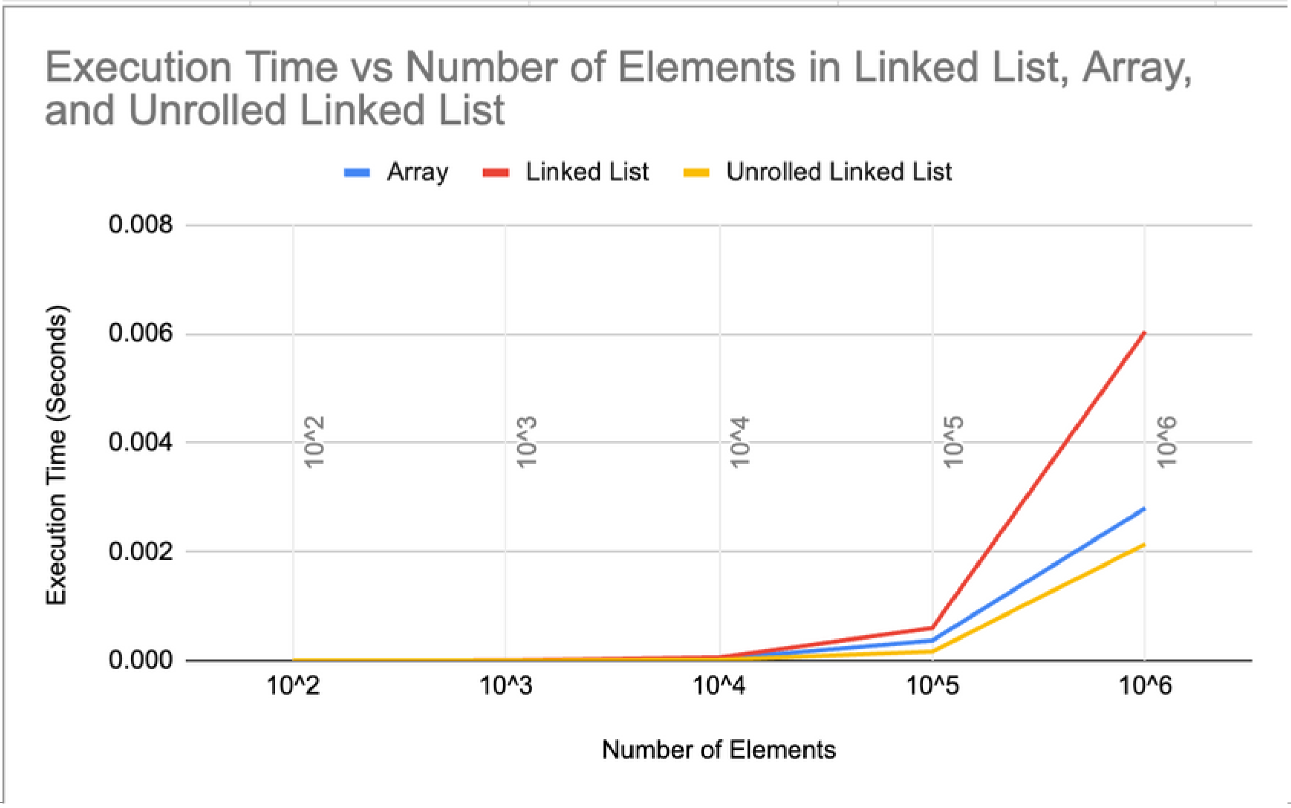
Both the array and the linked list have time complexities of O(n), yet they have different performance when n increases. This is because an array is stored contiguously in RAM and all elements are adjacent to each other. Locality is the idea that a computer will access the same section of ram multiple times, which increases performance. The reason it increases performance is because the CPU can cache the elements since they are next to each other in memory, allowing for quicker access. For example if there is an array with 3 elements, the memory addresses could be 0x0001 to 0x0003 (addresses were picked arbitrarily for example). This means that the array has good locality. When using a linked list, the pointers point to addresses that are not contiguously adjacent, meaning reduced locality, which results in less efficient performance times, which is shown in the graph.

2 Task 2: Locality Improved Linked List

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



Execution Time (Array)	Execution Time (Linked List)	Execution Time (Unrolled Linked List)	Elements
0.000001	0.000002	0.000001	10^2
0.000004	0.000006	0.000002	10^3
0.000032	0.000059	0.000015	10^4
0.000371	0.0006	0.000168	10^5
0.0028	0.006043	0.002141	10^6



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 a unrolled linked list is O(n) where n is the number of elements. This is because we are still iterating through each element only once, just in a different structure. As shown in the graph above it has better performance than the regular linked list and the array. This is due to the fact that each node contains an array with a predetermined amount of elements depending on the block size, which as previously stated, a regular array has better locality. This is the reason for the overall performance increase when iterating through the unrolled linked list. For example, if the block size is 100 and there is 10,000 elements, then there would only be 100 nodes, each with an array of 100 elements. This improves the overall locality because it reduces the amount of nodes in the linked list and increases the number of arrays.

The End