

\_\_\_\_\_

|              |   |
|--------------|---|
| First name   | Kevin   |
| Last name    | Wittlinger  |
| Collaborator | None  |
| Pledge       | I pledge may 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.

|   | A               | B       | C                     | D                    |
|---|-----------------|---------|-----------------------|----------------------|
| 1 | length          | Length  | Linked List Exec Time | Array List Exec Time |
| 2 | 10 <sup>2</sup> | 100     | 0.000002              | 0.000001             |
| 3 | 10 <sup>3</sup> | 1000    | 0.000009              | 0.000001             |
| 4 | 10 <sup>4</sup> | 10000   | 0.000095              | 0.000001             |
| 5 | 10 <sup>5</sup> | 100000  | 0.000669              | 0.000001             |
| 6 | 10 <sup>6</sup> | 1000000 | 0.005151              | 0.000001             |

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**.

Even though both data structures have the same complexity of  $\mathcal{O}(n)$ , they still have different performances because Arrays tend to leverage better spatial locality, this makes accessing memory and getting values much more efficient than Linked lists that don't function well spatial locality. This is because Arrays store memory continuously and linked lists have to create separate nodes and point to nodes that they want to access creating a more scattered pathway for the computer to compute.

## 2. Task 2: Locality Improved Linked List

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



| length  | Linked list execution time | Array execution time | Unrolled Linked List execution time |
|---------|----------------------------|----------------------|-------------------------------------|
| 100     | 0.000004                   | 0.000001             | 0.000001                            |
| 1000    | 0.000008                   | 0.000001             | 0.000006                            |
| 10000   | 0.00011                    | 0.000001             | 0.000044                            |
| 100000  | 0.000621                   | 0.000003             | 0.000298                            |
| 1000000 | 0.005101                   | 0.000001             | 0.001603                            |



execution 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 is  $O(n)$ . Unrolled linked lists improve efficiency of traversal by putting more elements within a singular node. With this clumping you would be able to access nearby elements in memory that had already been loaded into the cache to make a faster method of travel. This process is enhancing the Spatial Locality as no longer is the traversal filled with jumps which could cause cache misses.

The End 🖐