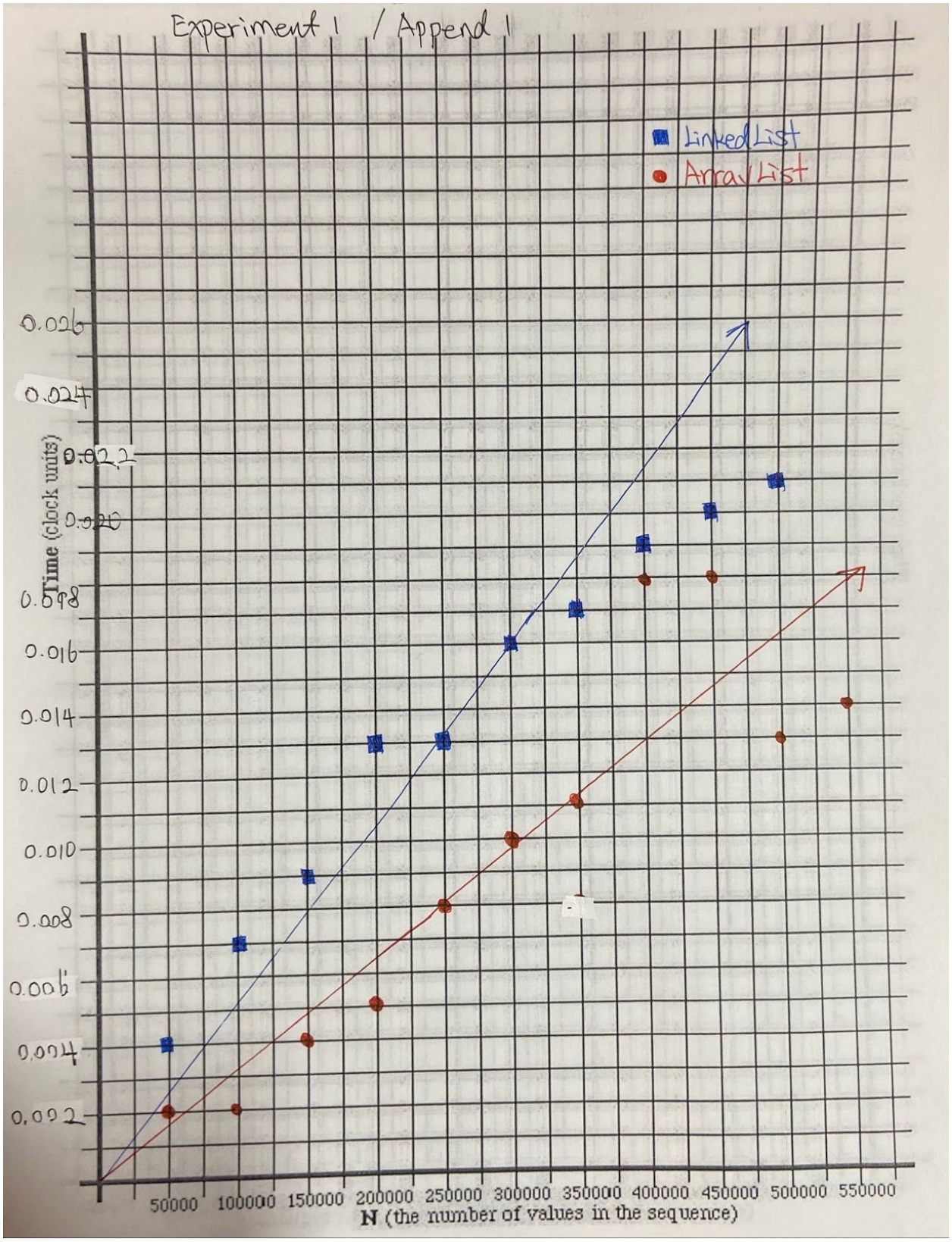
Experiment 1 -- Append 1

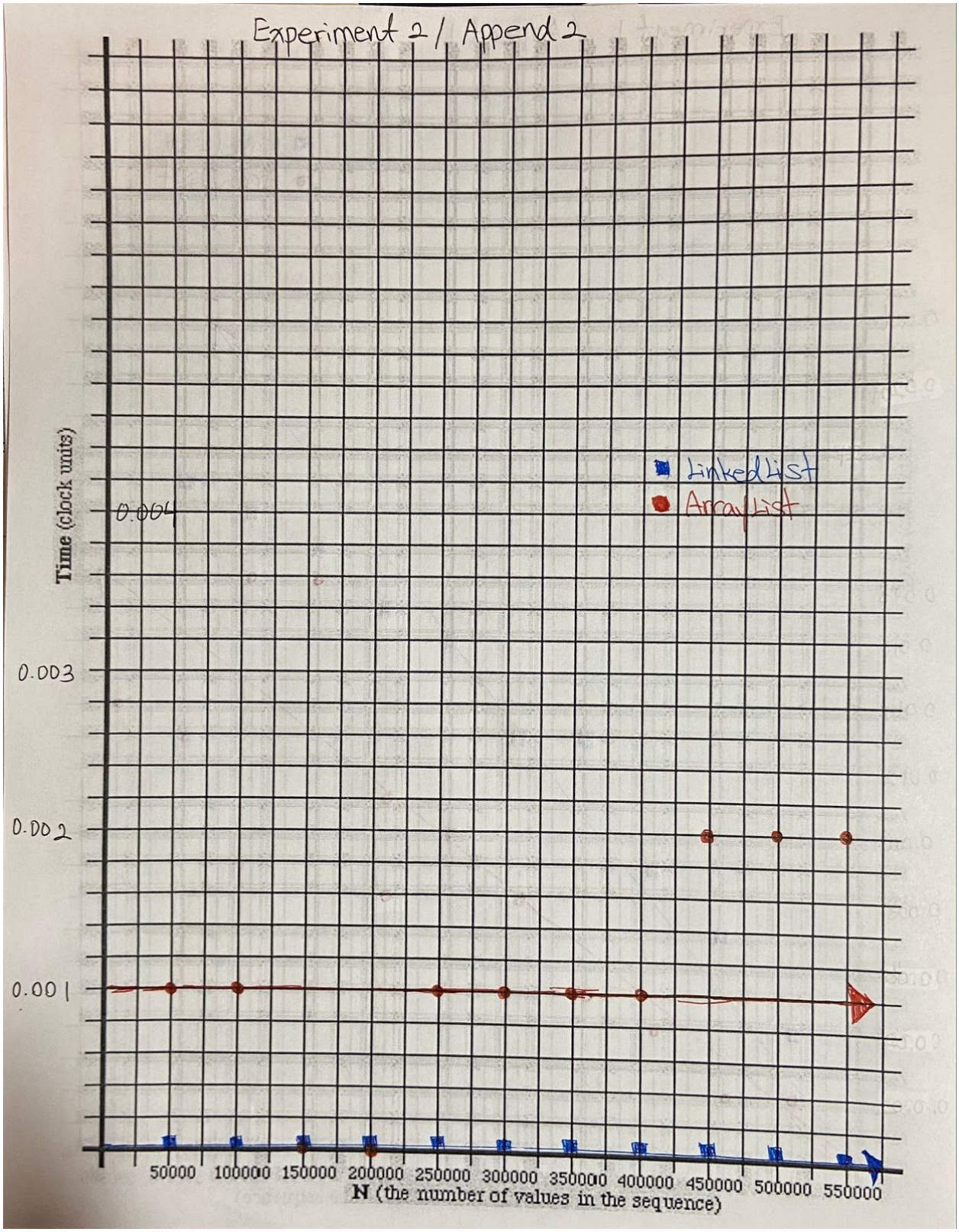


The running time for LinkedList in Append 1 (Experiment 1:Appending n Values to an Empty Container) is O(N), linear. "`aLinkedList.add(i);`" take constant time to add value. For each value of `i` in the range (0 … N-1) need to append to the end of the list. Append `i` is adding N times which taking linear time to run the code. O(1) * N = O(N)

The running time for ArrayList Append 1 (Experiment 1:Appending n Values to an Empty Container) is O(N), linear. "`anArrayList.add(i);`" take constant time to add value and the size of `anArrayList` affect the running time. . For each value of `i` in the range (0 … N-1) need to append to the end of the list. As the size of N increases, the time is increasing in linear.

ArrayList was not required to extend the memory, which means the use of `add()` method is ArrayList will faster than LinkedList.

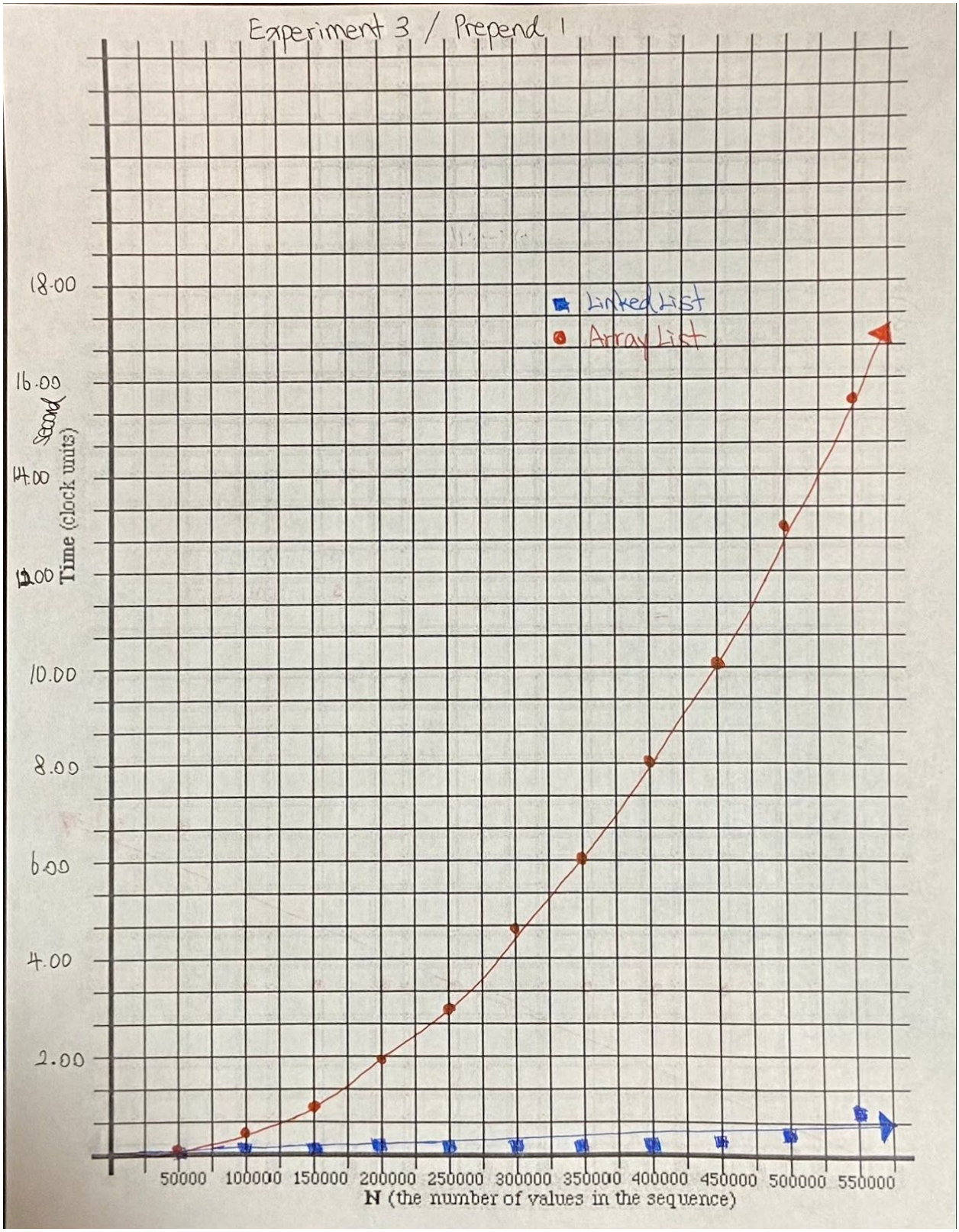
Experiment 2 -- Append 2



The running time for LinkedList in Append 2 (Experiment 2: Appending n Values to an Empty Container) is O(1), constant. The time for append 1 to the end of the list is constant, and it was independent of the size of the list.

The running time for ArrayList in Append 2 (Experiment 2: Appending n Values to an Empty Container) is O(1), constant. The time for append 1 to the end of the `ArrayList` is constant, and it was independent of the size of the list.

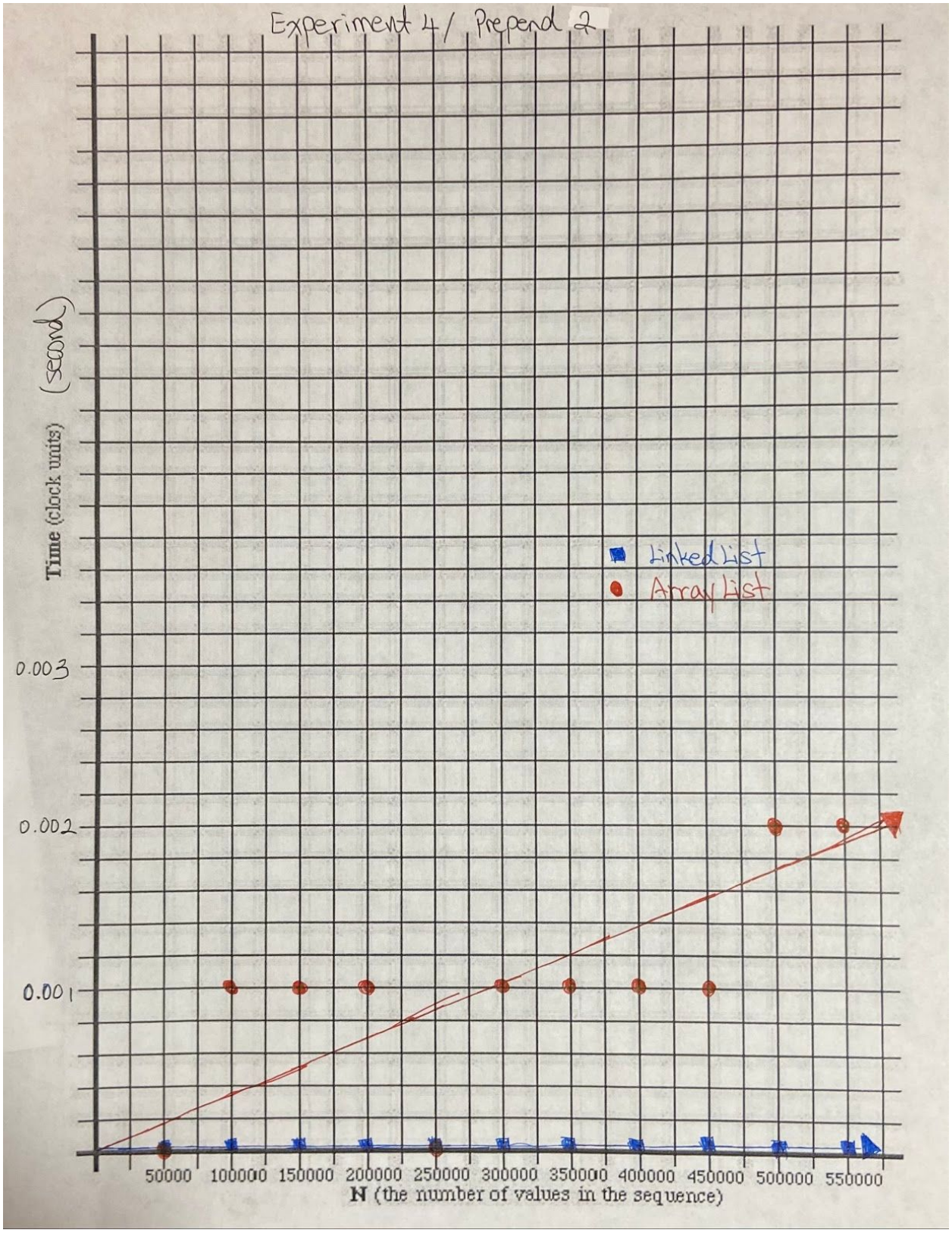
Experiment 3 -- Prepend 1



The running time for LinkedList in Prepend 1 (Experiment 3: Prepending n Values to an Empty Container) is O(N), linear. For each value `i` in the range (0 … N-1) need to insert 1 at the beginning of the LinkedList. The running time is based on the sizes of N, if N is increased, then the time was increasing in linear.

The running time for ArrayList in Prepend 1 (Experiment 4: Prepending n Values to an Empty Container) is O(N^2), quadratic. For each value `i` in the range (0 … N-1) need to insert 1 at the beginning of the ArrayList. When inserting 1 at the beginning of the `ArrayList`, it needs to extend the size of the list, which is O(N). To run N times is O(N). O(N) * O(N) = O(N^2)

Experiment 4 -- Prepend 2



The running time for LinkedList in Predend 2 (Experiment 4: Pretending 1 Value to a Container of Size n) is O(1), constant. Inserting 1 at the beginning of the `LinkedList` takes constant time and it was independent of the size of the list.

The running time for ArrayList in Predend 2 (Experiment 4: Pretending 1 Value to a Container of Size n) is O(N), linear. Inserting 1 at the beginning of the ArrayList needs to copy all the value in the list and it depended on the size of the list. As the size of the list increasing, the running time is increasing in linear.