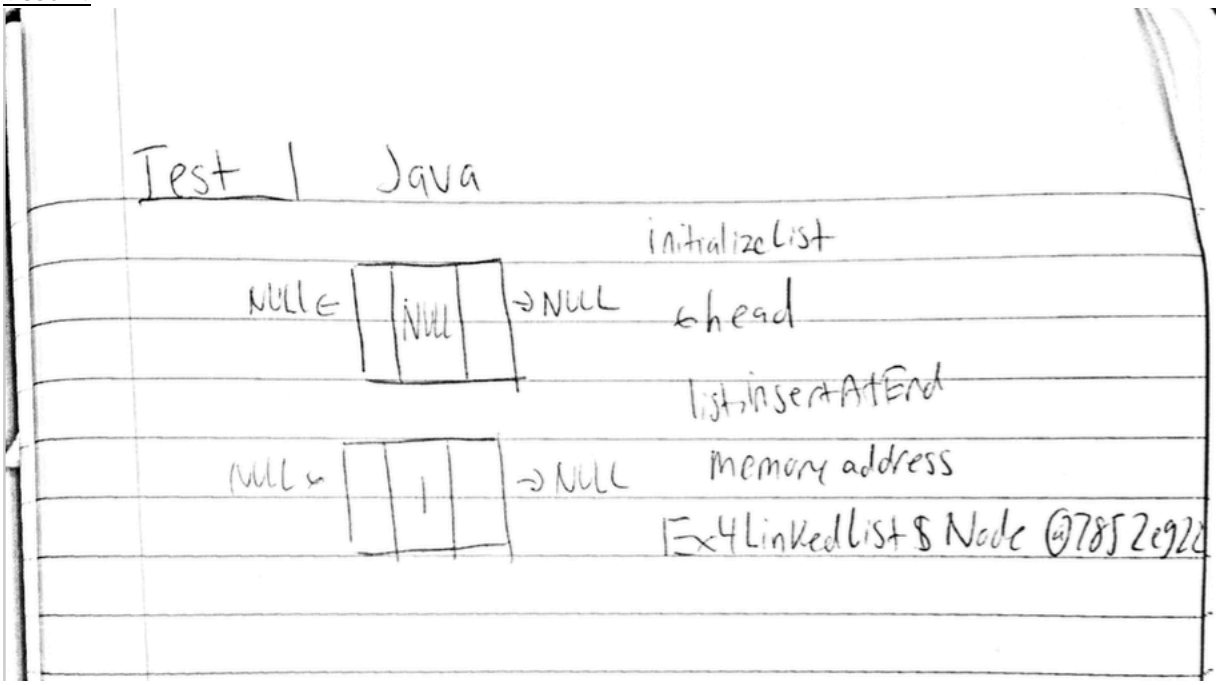
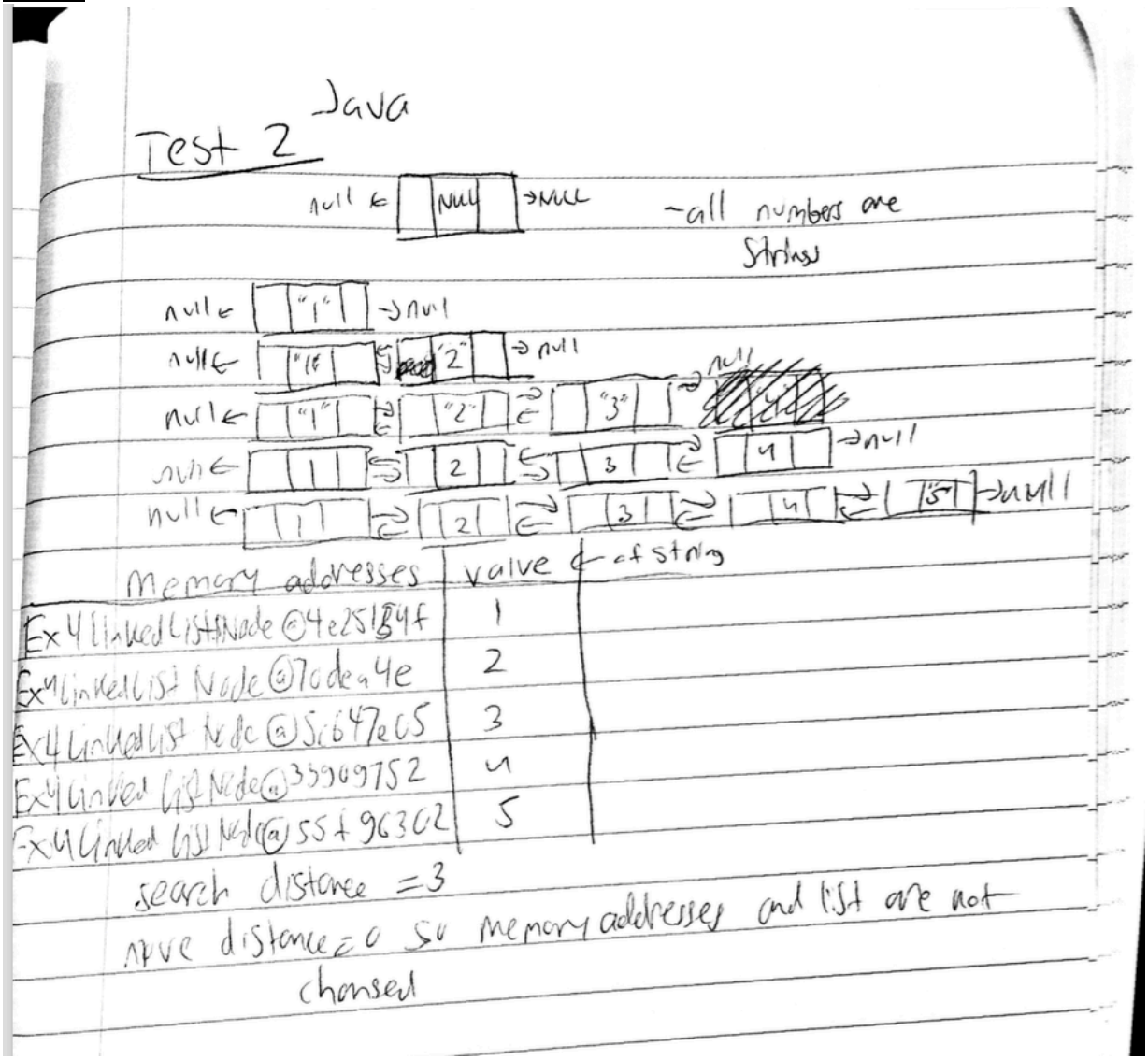
Exercise 4: PDF drawings of the list at each stage of insertion and list adjustment.
(Linked List in Java)
Test 1:



Test 2:


Test 2
Java
-all numbers are Strings
memory addresses
value of string
search distance = 3
save distance = 0 so memory addresses and list are not changed

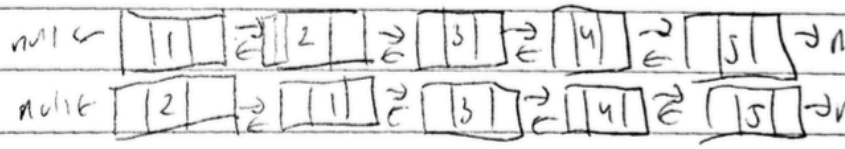
Test 3:

Java

Test 3 * Same as Test 2 until altered ~~except~~ memory addresses

Value	Memory Address
1	Ex4LinkedList$Node@3d4eac69
2	Ex4LinkedList$Node@42a57993
3	Ex4LinkedList$Node@75b84c92
4	Ex4LinkedList$Node@6bc7c054
5	Ex4LinkedList$Node@232204a1

position = 2 so value found is 2
moveDistance = 5



Memory address of the 2nd node is swapped with the memory address of the 1st node

Average search depth for a list of 10 elements using various values of moveDistance in the range of 1 to 10.

Using a list with 10 elements, 10 as the number of trials and values of moveDistance from 1 to 7, here are the values for average search distance:

Move Distance	Average Search Distance
1	7.2
2	6.4
3	4.7
4	5.7
5	5.5
6	6.8
7	5.3

Code to prove above: (nearly identical to the values from the C version)

```
10 elements
  Listsize=10 numTrials=10 moveDistance=1 avgSearchDistance=7.2
10 elements
  Listsize=10 numTrials=10 moveDistance=2 avgSearchDistance=6.4
10 elements
  Listsize=10 numTrials=10 moveDistance=3 avgSearchDistance=4.7
10 elements
  Listsize=10 numTrials=10 moveDistance=4 avgSearchDistance=5.7
10 elements
  Listsize=10 numTrials=10 moveDistance=5 avgSearchDistance=5.5
10 elements
  Listsize=10 numTrials=10 moveDistance=6 avgSearchDistance=6.8
10 elements
  Listsize=10 numTrials=10 moveDistance=7 avgSearchDistance=5.3
```