Assignment 2

I have implemented the assignment in the following 4 classes :

1. BubbleSort.java
2. DoublyList.java
3. Node.java
4. SingleList.java

The Node.java class is the base class and DoublyList.java and SingleList.java uses that class to implement doubly linked list and single linked list respectively. I have included the method to bubblesort in both these classes. The method first checks if the list is empty or if it has single element. The BubbleSort.java class generates random numbers and inserts them in the 2 types of lists and then later calls the bubblesort method present in these 2 classes.

In bubblesort method, I am going through the length of the linked list to check if the element at nth is greater than the element at n+1th , if it is I am exchanging that in the swap function, which puts the value in a temporary variable and then changes how the links are linking the elements.

I am also pasting a runtime example output which was generated when I ran it one time. As you can see it first lists all the elements and then prints all the elements in sorted order with the number of exchanges, traversals and node comparisons made. The comparisons and exchanges are higher if the list is in decreasing order, if all of the list is in decreasing order, that would be the highest number of exchanges and node comparisons. A linked list which is in decreasing order from the start to finish will have the worst time case complexity of O(n^2) to sort according to bubblesort.

In this example, the single linked list has 51 random elements. These 51 random elements are sorted in 1938 comaprisons and 520 exchanges. The traversals made were 38.

the Doubly linked list has 6 random elements. These 6 random elements are sorted in 30 comaprisons and 11 exchanges. The traversals made were

5.

Example output :

```
Size of the Single list is 51
The singly linked list is
 1. 28354 2. 9352 3. 16290 4. 30475 5. 27043 6. 25669 7. 9110 8.
15449 9. 12869 10. 24225 11. 38720 12. 11308 13. 11554 14. 44482
15. 2550 16. 6566 17. 2542 18. 43640 19. 38640 20. 24393 21.
43968 22. 45950 23. 29139 24. 41031 25. 35853 26. 4316 27. 6859
28. 37217 29. 37315 30. 18426 31. 37765 32. 26240 33. 37279 34.
28606 35. 39894 36. 40386 37. 2443 38. 4342 39. 48747 40. 48284
41. 31383 42. 42754 43. 23039 44. 33055 45. 7443 46. 34744 47.
44962 48. 48517 49. 45015 50. 21835 51. 15084
Number of exchanges made are 520
Number of node comparisons made are 1938
Number of traversals made are 38
The sorted singly linked list is
 1. 2443 2. 2542 3. 2550 4. 4316 5. 4342 6. 6566 7. 6859 8. 7443
9. 9110 10. 9352 11. 11308 12. 11554 13. 12869 14. 15084 15.
15449 16. 16290 17. 18426 18. 21835 19. 23039 20. 24225 21. 24393
22. 25669 23. 26240 24. 27043 25. 28354 26. 28606 27. 29139 28.
30475 29. 31383 30. 33055 31. 34744 32. 35853 33. 37217 34. 37279
35. 37315 36. 37765 37. 38640 38. 38720 39. 39894 40. 40386 41.
41031 42. 42754 43. 43640 44. 43968 45. 44482 46. 44962 47. 45015
48. 45950 49. 48284 50. 48517 51. 48747
-----------------------------------------------------------
Size of the doubly linked list is 6
The doubly linked list is
 1. 29207 2. 37511 3. 23054 4. 37414 5. 9312 6. 18458
Number of exchanges made are 11
Number of node comparisons made are 30
Number of traversals made are 5
The sorted doubly linked list is
 1. 9312 2. 18458 3. 23054 4. 29207 5. 37414 6. 37511
```