**Practical 02**

***Sorting Algorithms – Selection Sort and Insertion Sort***

Selection Sort and Insertion Sort are quadratic sorting algorithms that operate in the order of O(n2), The following practical will help you grasp how these algorithms are implemented and how the abstract concept of these sorting algorithms can differ from their concrete implementations.

**Consider the following array of Integer values.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **22** | **11** | **34** | **-5** | **3** | **40** | **9** | **16** | **6** |

1. Write the contents of the array for 5 passes of the outermost loop of insertion sort.

**Pass 1**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **11** | **22** | **34** | **-5** | **3** | **40** | **9** | **16** | **6** |

**Pass 2**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **-5** | **11** | **22** | **34** | **3** | **40** | **9** | **16** | **6** |

**Pass 3**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **-5** | **3** | **11** | **22** | **34** | **40** | **9** | **16** | **6** |

**Pass 4**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **-5** | **3** | **9** | **11** | **22** | **34** | **40** | **16** | **6** |

**Pass 5**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **-5** | **3** | **9** | **11** | **16** | **22** | **34** | **40** | **6** |

1. Write the contents of the array for 4 passes of the outermost loop of selection sort.

**Pass 1**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **-5** | **11** | **34** | **22** | **3** | **40** | **9** | **16** | **6** |

**Pass 2**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **-5** | **3** | **34** | **22** | **11** | **40** | **9** | **16** | **6** |

**Pass 3**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **-5** | **3** | **6** | **22** | **11** | **40** | **9** | **16** | **34** |

**Pass 4**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **-5** | **3** | **6** | **9** | **11** | **40** | **22** | **16** | **34** |

1. Create a Selection Sort class that has a sorting method that will accept an array as an argument and will return a sorted array. **Completed in SelectionSort.java**
2. Create an Insertion Sort class that has a sorting method that will accept an array as an argument and will return a sorted array. **Completed in InsertionSort.java**
3. Write a piece of code that will propagate a static array of 10000 random elements. **Completed in Q5.java**
4. Write a piece of code that will propagate a static array of 10000 sorted elements. **Completed in Q6.java**
5. Write a piece of code that will propagate a static array of 10000 inversely sorted (sorted z – a). **Completed in Q7.java**
6. Implement an array list, populating it similar to the arrays in 5, 6 and 7 above. **Completed in Q8.java**
7. Rewrite the Insertion Sort and Selection Sort algorithms to sort the array lists. **Completed in SelectionSort.java, InsertionSort.java**
8. Does changing the data structure from a static array to an array list have an impact on the overall efficiency of the algorithm?   
   **-It does affect the efficiency of the algorithm by a little bit, however you need to change your code to suit an ArrayList.**
9. Create a print method that will print both the initial array and the sorted version. **Completed in Q8.java**
10. Write a print method that will validate your results in steps 1 and 2 above. **Completed in Q12p1.java, Q12p2.java**
11. Write code that will record the number of comparisons that the Selection and Insertion Sort algorithms will carry out on the datasets implemented in steps 5, 6, and 7 above. **Completed in SelectionSort.java, InsertionSort.java**
12. Write code that will record the number of swap operations that the Selection and Insertion Sort algorithms will carry out on the datasets implemented in steps 5, 6, and 7 above. **Completed in SelectionSort.java, InsertionSort.java**