Program Structure and Algorithm

Assignment – 3

Name – Surbhi Soni

NUID - 002969578

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Random Order Array (ns) | Ordered Array(ns) | Partially ordered(ns) | Reverse-ordered(ns) |
| 1000 | 0.775821 | 0.004002 | 0.516064 | 1.424809 |
| 2000 | 2.897877 | 0.00762 | 2.122473 | 5.642333 |
| 4000 | 7.392344 | 0.009129 | 5.089738 | 13.654428 |
| 8000 | 27.535197 | 0.014684 | 21.437602 | 57.109556 |
| 16000 | 115.595747 | 0.177816 | 84.366844 | 211.269149 |

Conclusion:

The conclusion from this output is that the running time of the insertion sort algorithm is affected by the initial ordering of the elements in the array. For arrays with a random ordering, the running time is significantly higher compared to arrays with an ordered or partially ordered ordering. Arrays with reverse-ordered orderings also have a higher running time compared to arrays with an ordered or partially ordered ordering.

Furthermore, the running time increases as the size of the array increases, regardless of the initial ordering. This suggests that the insertion sort algorithm is not the most efficient sorting algorithm for large arrays.

It can also be seen that for arrays with an ordered ordering, the running time is significantly lower compared to arrays with a partially ordered or reverse-ordered ordering. This indicates that the insertion sort algorithm performs best when the elements are already partially sorted.

Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generated

Text

Description automatically generated