|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Size | Quick (s) | Insertion (s) | Merge (s) | Selection (s) | Bubble (s) |
| 20,000 | 0.004 | 0.547 | 0.013 | 0.766 | 2.453 |
| 40,000 | 0.009 | 2.266 | 0.028 | 3.152 | 9.857 |
| 60,000 | 0.013 | 4.973 | 0.042 | 8.300 | 22.142 |
| 80,000 | 0.017 | 8.734 | 0.055 | 12.481 | 39.963 |
| 100,000 | 0.024 | 14.406 | 0.072 | 19.453 | 62.129 |

Overall, quick sort and merge sort are the best options for sorting large datasets efficiently, while bubble sort, insertion sort, and selection sort are not recommended for larger datasets. Additionally, it is important to consider the time complexity of an algorithm when selecting a sorting algorithm for a given dataset.

Algorithms with a 2nd-degree polynomial time complexity (insertion sort, bubble sort and selection sort) can be seen about quadrupling the time when the input size is doubled. Quick and Merge sort with complexity O(nlogn) can be seen to be far more efficient even as the input size is increasing, the increase in time taken is almost linear (as we have smaller array sizes).

## Screenshots:

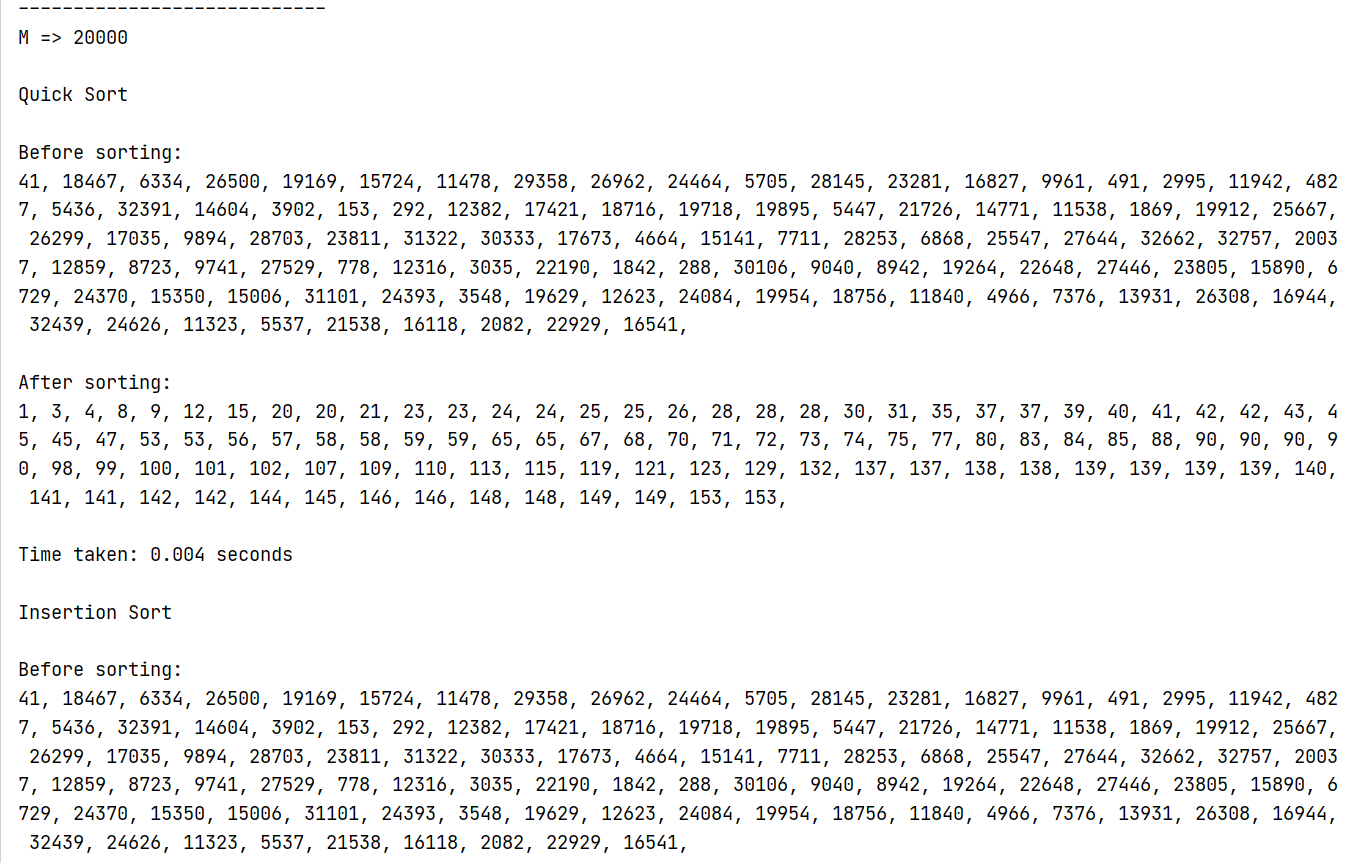


Figure 1: Beginning of program

A picture containing text

Description automatically generated

Figure 2: End of program