**Project 2**

The first element in the each order is comparisons, followed by number of movements followed by the time it took to Sort the particular order in a particular sorting method. The comparisons column has the total number of comparison it took for all the different orders. The Movements column has the total number of movements it took for all the different orders. The Total time column has the total time it took for all the different orders.

Inorder generates elements 0-n in ascending order. ReverseOrder generates elements 0-n in decending order. AlmostOrder generates elements 0-n in ascending order for 70% of n and generates random elements for the remaining of the array. Random generates elements 0-n Randomly using math.random.

I used 1 movement considering 1 swap process that is the whole process of putting a value into a temp variable and changing the values of the array. My time is In nanoseconds and it varies a lot due to that.

Main asks the user the order and the number of elements they want the array to be generated in. This uses Switch-case to generate the required array. Then it asks the user the sorting method they want to use to sort the array. This uses another Switch-Case which goes to the respective case accordingly and passes the array to the respective class and sorts the array. There is a variable keeping check of the lowest time it took for the sort which is determining the winning array. The Program loops until the user presses 0 when asked to end the program and then the program displays a goodbye message.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Experimental Results | InOrder | ReverseOrder | AlmostOrder | Random | Array Size | Comparisons | Movements | Total Time in nanoseconds |
| Insertion Sort | 9,0,13217 | 45,45,12503 | 11,2,2485 | 21,12,3603 | 10 | 89 | 59 | 31808 |
| Selection Sort | 45,9,16047 | 45,9,6581 | 45,9,4658 | 45,9,4343 | 10 | 180 | 36 | 31629 |
| Quick Sort | 45,54,33329 | 45,29,14429 | 45,54,11339 | 21,17,9532 | 10 | 156 | 154 | 68629 |
| Merge Sort | 19,34,53166 | 15,34,12565 | 19,34,12195 | 22,34,13190 | 10 | 75 | 136 | 91116 |
| Heap Sort | 72,31,37224 | 54,22,10992 | 74,32,12401 | 70,30,23095 | 10 | 270 | 115 | 83712 |
| Radix Sort | 9,19,232755 | 9,20,61285 | 9,128,50590 | 9,122,58110 | 10 | 36 | 289 | 402740 |