Basic code to calculate and display the time taken for a sorting process to complete.

```
#include <chrono>
using namespace chrono;

auto start = high_resolution_clock::now();
        // Selection sorting process
auto stop = high_resolution_clock::now();

auto selectionSortDuration = duration_cast<microseconds>(stop - start);

cout << "    Selection Sort: " << selectionSortDuration.count() << endl << endl;
```

ARRAY SIZE: 1000, Integer Sorting

Selection Sort:	3113	2918	4808
Insertion Sort:	1464	1705	1944
Quick Sort:	1664	1658	2048
Merge Sort:	2123	2043	4670
Heap Sort:	2723	2430	2411

ARRAY SIZE: 5000, Integer Sorting

Selection Sort:	40746	45969	39125
Insertion Sort:	19194	27594	18092
Quick Sort:	6124	7093	4295
Merge Sort:	8151	8113	6999
Heap Sort:	7926	10892	10073

ARRAY SIZE: 10000, Integer Sorting

Selection Sort:	166847	144809	150194
Insertion Sort:	80830	78779	74179
Quick Sort:	8869	10390	10246
Merge Sort:	12331	16065	14055
Heap Sort:	19858	13309	22458

ARRAY SIZE: 50000, Integer Sorting

Selection Sort:	3419079	3497861	3446979
Insertion Sort:	1869423	1893290	1881408
Quick Sort:	59747	40267	62214
Merge Sort:	77697	69965	78512
Heap Sort:	96508	88231	94943

ARRAY SIZE: 1000, Stock Sorting

Selection Sort:	30929	24746	28569
Insertion Sort:	331621	215819	379631
Quick Sort:	63577	43417	58593
Merge Sort:	161604	109009	165094
Heap Sort:	82585	52221	85294

ARRAY SIZE: 5000, Stock Sorting

Selection Sort:	480789	486078	466186
Insertion Sort:	6630123	6522059	6722095
Quick Sort:	249379	345591	318812
Merge Sort:	898163	792488	792292
Heap Sort:	378431	431101	404756

ARRAY SIZE: 10000, Stock Sorting

Selection Sort:	1897632	1819203	1913252
Insertion Sort:	23759807	24366901	24794376
Quick Sort:	540993	518055	497080
Merge Sort:	1424669	1416681	1441347
Heap Sort:	790203	754458	714833

ARRAY SIZE: 50000, Stock Sorting

Selection Sort:	44167932	46745817	545168072
Insertion Sort:	659105441	654152092	545168072
Quick Sort:	3375726	3960633	2849159
Merge Sort:	9651087	9924754	8287146
Heap Sort:	4995249	5108199	4120589