

## INTRO:

We are implementing four header files called [BubbleSort.h - HeapSort.h - MergeSort.h - SelectionSort.h] and one main cpp file called SortTimeDriver.cpp

They all contain functions that use a template that uses the class ItemType

---

## DESCRIPTION OF FILES:

1. **BubbleSort.h**

This header file contains everything related to the bubble sort algorithm

2. **HeapSort**

This header file contains everything related to the heap sort algorithm

3. **MergeSort**

This header file contains everything related to the merge sort algorithm

4. **SelectionSort**

This header file contains everything related to the selection sort algorithm

5. **SortTimeDriver**

This cpp file contains the main function and calls upon all the previous files and functions

---

## TEST RUNS

Test run 1:

Sample size: 50 ints

SelectionSort execution time in seconds: 4.8e-06

BubbleSort execution time in seconds: 7.1e-06

MergeSort execution time in seconds: 4.8e-06

HeapSort execution time in seconds: 8e-07

Test run 2:

Sample size: 500 ints

SelectionSort execution time in seconds: 0.0002891

BubbleSort execution time in seconds: 0.0005407

MergeSort execution time in seconds: 5.93e-05

HeapSort execution time in seconds: 6e-06

Test run 3:

Sample size: 5000 ints

SelectionSort execution time in seconds: 0.026402

BubbleSort execution time in seconds: 0.0558917

MergeSort execution time in seconds: 0.0015536

HeapSort execution time in seconds: 5.15e-05

Test run 4:

Sample size: 50000 ints

SelectionSort execution time in seconds: 2.60405

BubbleSort execution time in seconds: 7.40761

MergeSort execution time in seconds: 0.0818791

HeapSort execution time in seconds: N/A

---

## OBSERVATIONS

After observing all test runs, MergeSort seems to be the fastest sorting algorithm out of these 4

HeapSort has a size limit so it needs to be increased to handle more samples