

# Programming Assignment 4

Due 30 SEP @ 11:59pm

Write a program that implements the merge sort algorithm. A template is provided that will read in an array of unsorted Comparable items, pass to the sort method, and print out whether or not the items are sorted and the time that it took to sort the items. The mergeSort and merge methods are static and no class/objects are required.

When you have your implementation complete, run a series of experiments where you double the input size each time. Start at 1,000,000 and stop at 32,000,000. The template will print out the time. Write up an analysis of the experiment and conjecture what the asymptotic running time (i.e. Big-Oh) is based on the results. The analysis must include the results of the timing experiments. The analysis should be short, one or two paragraphs.

## Grading Notes

You must:

- Use the template provided for you
- Have a style (indentation, good variable names, etc.)
- Comment your code well (no need to over do it, just do it well)

You may not:

- Make your program part of a package.
- Use *code* from anywhere except your own brain.
  - This includes Java Arrays.sort(...) methods

Submission Instructions:

- Name a folder with your gmu username
- Put your java files in the folder (but not your .class)
- Zip the folder (not just the files) and name the zip "username-pa2.zip"
- Submit to blackboard

## Grading Rubric

No Credit:

- Non-submitted assignments
- Late assignments
- Non-compiling assignments
- Non-independent work

1pt	Submission Format
1pt	Style and Comments
3pts	mergeSort method
3pts	merge method
2pts	Analysis

### Example Run

```
> java MergeSort input.txt  
true  
Time=0.633774ms
```

### Empirical Runs

```
> java MergeSort 1000000  
true  
Time=???ms  
> java MergeSort 2000000  
true  
Time=???ms  
> java MergeSort 4000000  
true  
Time=???ms  
> java MergeSort 8000000  
true  
Time=???ms  
> java MergeSort 16000000  
true  
Time=???ms  
> java MergeSort 32000000  
true  
Time=???ms
```