

# CPSC 331 — Assignment #3

## Implementing a Sorting Algorithm

### About This Assignment

This can be completed by groups of up three students. It is due by 11:59pm on Monday, June 17.

1. A file

`Array.java`

is now available. This provides a bounded array that works well with generic code. Please *do not modify* this file when working on this assignment.

A much less complete file

`ArrayUtils.java`

is now available as well.

Complete the implementation of the method with signature

```
public void sort (Array<T> A)
```

in this file.

For full marks — or even very generous *part marks* — the algorithm that you implement, must have the following properties.

- The number of steps used to sort an Array with length  $n$  must be in  $O(n \log n)$  in the **worst case** — not just “most of the time.”
- The algorithm must *sort its input array in place*, using  $O(\log n)$  additional storage — including the size of a call stack whenever a recursive method is being used.

A small number of **bonus marks** will be given if you implement a version of the algorithm that only uses a **constant** amount of additional storage — provided that you sketch brief proofs of the correctness and efficiency of the algorithm that you have chosen.

2. Write a report describing the algorithm you selected, and explaining why you chose it.

If this is different from an algorithm that was provided in the course lecture notes then sketches of proofs of the correctness and efficiency of the algorithm should be included in your answer to this question.

If you have simply implemented an algorithm from class then report will probably be quite brief.