

Lab Tasks 9

No submission.

1. Download the assignment project from <http://www.comp.nus.edu.sg/~cs1102s/java/Lab9.zip>.

Complete the class `sorting.BubbleSort.java` to implement the following sorting algorithm:

Input: Unsorted array a of size n

Algorithm: Complete $n - 1$ passes. In each pass k , run through the elements at position 1 to $n - k$, comparing the element with its neighbor to the right. If they are in the wrong order, swap them.

Optimization: Terminate the whole sorting if you complete a pass that leads to no swap.

What is the best-case and worst case complexity of bubble sort?

2. Complete the class `sorting.ShellHibbard.java` to implement Shellsort using Hibbard's increments.
3. Complete the class `sorting.RadixSort.java` to implement the following sorting algorithm called *radix sort*. You can assume that the objects to be sorted are strings whose characters are upper-case letters from 'A' to 'Z', such as HELLO.

The algorithm runs in phases, one phase for each position in the given strings, starting from the least significant character. In each phase, the strings are sorted into groups according to the position under consideration. It is important to make sure that the ordering with respect to a phase is not disturbed in subsequent phases.

The algorithm in more detail:

```
radixSort(theArray: array of strings)
  identify the size n of theArray and the longest string size d
  for (j = d down to 1) {
    Initialize 26 groups to empty
    Initialize a counter for each group to 0
    for (i = 0 through n - 1) {
      k = j-th letter of theArray[i]
      Place theArray[i] at the end of group k
      Increase k-th counter by 1
    }
    Replace the strings in theArray with all the strings in group 0,
    followed by the strings in group 1, and so on
  }
```

What is the worst-case complexity of radix sort?