

INF102- Compulsory Assignment 1

1 – Simple calculator

Code has been written in the file SimpleCalculator.java, which has been attached. It works for single digit numbers and multiple digit numbers

2 – Triplicates in four lists

Code has been written in the file Triplicates.java, which has been attached.

3 – Quicksort, empirically

The code has been attached in the file QuickEmpirically.java. The testing is shown in the attached PDF, which shows the testing has been done 10 times, and the graphs were all done with the mean of those.

This was finished before the note about compiler optimization was added, and one of the group teachers said that it was fine to just deliver it, that I didn't have to do it all over again.

4 – Countsort

The code has been attached in the file Countsort.java, and the numbers and graph has been attached as the file Countsort.pdf. The runtime analysis is also in the pdf.

5 – Exam exercise

Question:

This exercise is built on the previous exercise. Explain how the algorithm Countsort works, and also illustrate it by drawing the arrays filled with the numbers.

Answer:

Countsort works by first finding the largest and smallest number in the given array. Then it creates a new array, commonly called a bucket, the size of the span from 0 to the largest number. This array is then filled in by incrementing the slot in the array by 1 each time the number appears in the original array. Example, if the number 5 appears 2 times in the original array, the temporary array at spot number 5 would be 2 ($\text{temp}[5] = 2$). After this process is complete, the algorithm fills in a new array the size of the original array by decrementing the bucket until 0 at the given spot. Illustrated below.

Original array, unsorted

4	5	2	3	4	4	0	5	3	5
---	---	---	---	---	---	---	---	---	---

The "bucket". On top: Array slot, on the bottom: The number of times that number appears.

0	1	2	3	4	5
1	0	1	2	3	3

The finished array, all sorted

0	2	3	3	4	4	4	5	5	5
---	---	---	---	---	---	---	---	---	---

