CSE 204
Offline 7

Problem: Sorting Algorithms
Objective: Comparison of Merge Sort and Quicksort

Task 1: Implementation

1. You have to implement Merge Sort and Quicksort
2. You are given a cpp file **sortarray.cpp**
3. Generate arrays with best case, average case and worst case scenario
4. Populate the array of size n by generating random integers
5. Apply merge sort and quicksort to sort the array
6. Record the time to accomplish each sorting
7. Finally print the array
8. Each generation process should generate an entirely new array
9. Your implementation must be memory efficient

Task 2: Statistics generation

10. Create another file **statistics.cpp**
11. Vary array size from 10 to 1000000. You can increase the upper range if that gives you better statistics.
12. Generate best case, average case and worst case scenario, sort them by merge sort and quick sort and record the timing in the following table.
13. For example: You want to get timing for n=10, best case with merge sorting. Generate the scenario multiple times and take the average sorting time. Record only the average sorting time into the cell.
14. Plot running time of both the sorting algorithm against the input array size n for best, worst and average case.

	n =	10	100	1000	10000	100000	1000000
Case	**Sort**						
best	merge						
	quick						
worst	merge						
	quick						
average	merge						
	quick						

Note:

1. Code in C++
2. You have to submit the codes and a report containing complexity analysis, machine configuration, table and plots

Submission

1. Create an empty folder named to your `student_id` (e.g. 1705001)
2. Put all the source code (.cpp) files and reports in that folder
3. Zip that folder. It should give you `student_id.zip`
4. Submit the zip file to moodle

Prepare to sit for an online.