COMPUTER SCIENCE TRIPOS Part IA – 2018 – Paper 1

7 Algorithms (RKH/DJW)

This question considers sorting arrays of numbers.

(*a*) Mergesort can be implemented as a conventional two-way mergesort or a three-way mergesort. The latter splits the input into three and applies three-way mergesort recursively to each segment.

(*i*) Derive an expression for the worst-case number of comparisons needed to merge *two* sorted arrays of length $\frac{n}{2}$. [2 marks]

(*ii*) Derive an expression for the worst-case number of comparisons needed to merge *three* sorted arrays of length $\frac{n}{3}$. Consider merging three arrays at once as well as merging in pairs. [5 marks]

(*iii*) Using your answers for Parts (*a*)(*i*) and (*a*)(*ii*) and solving suitable recurrence relations, find expressions for the total number of comparisons needed for two-way and three-way mergesort. [6 marks]

(*iv*) Assuming comparisons to be the dominant cost, would you expect two-way or three-way mergesort to perform faster on an arbitrary array? [2 marks]

(*b*) Consider a large dataset of numbers between 0.0 and 1.0 drawn from some known non-uniform distribution. Describe a sorting algorithm that would be expected to perform better than the above mergesort on this data. What complexity would your algorithm have in terms of space and time in the worst and average cases? [5 marks]