COMP3711: Design and Analysis of Algorithms

Tutorial 5

HKUST

Question 1

Design an $O(n \log k)$-time algorithm to merge k sorted lists into one sorted list by making use of priority queues, where n is the total number of elements in all input lists. Note that each sorted list may contain different number of elements.

Question 2

Given n/k lists where each list contain k elements and the elements in list $i-1$ are less than the elements in list i for $i=1$ to n/k. Show that $\Omega(n \log k)$ is the lower bound for any comparison-based sorting algorithm to sort the n/k lists into one sorted list with n elements. Note that you should not simply combine the lower bounds for the individual lists.

Question 3

Give an array of m positive integers, where different integers may have different number of bits, but the total number of bits over all the integers in the array is n. Show how to sort the array in O(n) time.

Note that running radix sort directly won't work, as the maximum integer may be as large as 2^n, so radix sort would take $O(m \log_m 2^n) = O(mn/\log m)$ time.