1. You are given \(n\) intervals \(\ell_i = [a_i, b_i]\) on the real line, where \(a_i \leq b_i\) and \(1 \leq i \leq n\). Design and analyze an algorithm that computes the measure of this set of intervals, that is, the total length of \(\bigcup_{i=1}^{n} \ell_i\), in \(O(n \log n)\) time.

2. A sorting algorithm is called \textit{stable} if the relative order of records with the same key values is the same after the sorting.
   
   (a) Show that bubble sort is stable.
   
   (b) Show that heap sort is not stable.

3. Instead of a two-way split, one can devise a merge sort algorithm that splits a list into three sublists of roughly equal sizes, recursively sorts them, and then merges the three sorted lists into one sorted list.
   
   (a) Work out the details of this algorithm, and analyze its running time. For simplicity, to carry out the analysis, assume that the input size \(n\) is a power of 3.
   
   (b) Which algorithm performs fewer comparisons (as \(n \to \infty\)), the three-way merge sort, or the usual two-way merge sort? Explain the reason behind your answer. Note that for this exercise, you cannot totally ignore constants factors!

4. Suppose that an array \(A[1 \ldots n]\) is organized as a heap, with the smallest element at \(A[1]\). The operation \(\text{HeapDelete}(A, v)\) deletes the item in node \(v\) in the heap (where \(1 \leq v \leq n\)). Give an implementation of \(\text{HeapDelete}\) that runs in time \(O(\log n)\).

5. Give an \(O(n \log k)\)-time algorithm to merge \(k\) sorted lists into one sorted list, where \(n\) the the total number of elements in all the input lists.