

1. Show that the running time of the Merge Sort algorithm on an n -element sequence is $O(n \log n)$, even when n is not a power of two.
 2. Given two ordered sequences corresponding to sets A and B , write an algorithm to compute a sequence corresponding to the set $A \oplus B$
 3. Given an ordered sequence corresponding to a multiset A , find an ordered sequence corresponding to the set A . Also determine the running time of your algorithm.
 4. Describe a radix-sort method for lexicographically sorting a sequence S of triplets (k, l, m) , where k, l , and m are integers in the range $[0, N - 1]$ for $N \geq 2$.
 5. For what sort of input sequence(s) will the Quick Sort algorithm runs slowest, assuming that the algorithm chooses as the pivot element (a) the first element of the sequence, (b) the last element of the sequence.
 6. Consider the modification of the classical version of the Quick Sort algorithm so that instead of choosing the last (or first) element of the n -element sequence as the pivot element, we choose the $\lfloor n/2 \rfloor$ th element as the pivot. Describe the kind of sequence that would cause this Quick Sort to run as fast as possible.
 7. Suppose we are given a sequence of n elements, each of which is colored red or blue. Give an in-place method for ordering S so that all the blue
-

elements are listed before all the red elements.

8. Suppose we are given a sequence of n elements, each of which is colored red, blue, or yellow. Give an in-place method for ordering S so that all the blue elements are listed before all the red elements, and all the red elements are listed before all the yellow elements.
 9. Given an n -element sequence, write an algorithm to find the median of the sequence. What is the worst-case and expected time of your algorithm.
 10. Prove that Randomized Quick Sort runs in $O(n \log n)$.
-