CSCI 2300- Home Work 4

Q1. (20 points) Compute tight big-Oh bounds for the following recurrences:
a. $T(n) = 8T(n/4) + O(n)$
b. $T(n) = 2T(n/4) + O(\sqrt{n})$
c. $T(n) = T(n-4) + O(n^2)$
d. $T(n) = T(\sqrt{n}) + O(n)$

Q2. (10 points) Let A be an array of n integers, and let R be the range of values in A, i.e., $R = max(A) - min(A)$. Give an $O(n + R)$ time algorithm to sort all the values in A.

Q3. (15 points) Let A be an array of n distinct integers. Consider an algorithm to find the minimum value, where we pair up the elements, and retain the smaller of the values from each pair. This will result in an array of half the size (actually the resulting size will be $\lceil n/2 \rceil$. We can then recursively apply the same approach, until we get a final array with just two elements. We compare these two values and return the minimum of those values as the answer. Answer the following questions:
a. How many comparisons are done in the above algorithm in the worst case.
b. Show how to modify/extend this method to find the second smallest element.
c. Prove that we can find the second smallest element in $n + \lceil \log(n) \rceil - 2$ comparisons in the worst case.
Note that the above questions are not asking for the big-Oh complexity, but rather the (exact) number of comparisons in the worst case. For example, it is easy to find the 2nd smallest element in at most 2n comparisons, but that is $n + n$ which is larger than $n + \lceil \log(n) \rceil - 2$ comparisons.