Problem 1) Master Theorem application. Use the Master theorem to identify the runtime of these Recurrences. In your answer, clearly identify the terms $a$, $b$, and $f(n)$.

1a. (5 points) $T(n) = 4T(n/4) + n$
1b. (5 points) $T(n) = 2T(n/4) + n$
1c. (5 points) $T(n) = 3T(n/9) + n \log n$
1d. (5 points) $T(n) = 2T(\sqrt{n}) + n$

Problem 2) BucketSort – defined in Cormen Section 8.4

2a. (20 points) Using Cormen Figure 8.4 as a model, illustrate the operation of Bucket-Sort on the array $A = [.79, .13, .16, .64, .39, .20, .89, .53, .71, .42, .25, .19, .88, .54]$.

2b. (20 points) What is the worst-case running time for the Bucket-Sort algorithm? What simple change to the algorithm preserves its linear expected running time and makes its worst case running time $O(n \log n)$.

Problem 3) (20 points) Describe an algorithm to sort $n$ integers in the range $0$ to $n^2 - 1$ in $O(n)$ time? Hint: Think about how Radix Sort works.

Problem 4) (20 points) Given two unsorted arrays, $A$ and $B$. Describe, in pseudocode, an efficient algorithm to identify the elements that appear in both $A$ and $B$. If an element appears more than once in $A$ and more than once in $B$ it should appear more than once in the output of the algorithm. Full credit will be awarded for an algorithm that is $\Theta(n \log n)$. Algorithms that are $\Theta(n^2)$ or slower will be scored out of 15 points.