Question 1. [6 *marks*]

(a) (3 *marks*) Give pseudocode for the Bucket-Sort algorithm for sorting an array.

(b) (1 *mark*) Give a recurrence for the running time of Bucket-Sort. Use n for the size of the array and n_i to denote the number of items in bucket i.

(c) (2 *marks*) Use your recurrence in (b) to show that the expected running-time of Bucket-Sort is $\mathcal{O}(n)$. You may use the fact that $\mathrm{E}[n_i^2] = 2 - \frac{1}{n}$.

Question 2. [7 *marks*]

Recall the Select algorithm for finding the i^{th} smallest element in an array that uses the median-of-medians as pivot element. The following recurrence describes the running time of Select.

$$T(n) \leq T(\lceil \tfrac{n}{5} \rceil) + T(\tfrac{7n}{10} + 6) + \mathcal{O}(n).$$

(a) (4 *marks*) Explain in detail what the terms $T(\lceil \frac{n}{5} \rceil)$ and $T(\frac{7n}{10} + 6)$ in the above recurrence represent. (Be sure to also explain how $\lceil \frac{n}{5} \rceil$ and $\frac{7n}{10} + 6$ are obtained.)

(b) (3 *marks*) Use the substitution method to show that the running time of Select is $\mathcal{O}(n)$.

Question 3. [5 *marks*]

Consider the following pseudocode for Inorder-Tree-Walk:

```
Inorder-Tree-Walk(x)
if x ≠ NIL
    Inorder-Tree-Walk(x.left)
    print x.key
    Inorder-Tree-Walk(x.right)
```

Let $T(n)$ denote the time taken by Inorder-Tree-Walk when it is called on the root of an n-node tree. Complete the following steps to show that $T(n) = \Theta(n)$.

(a) (1 *mark*) Explain why is $T(n) = \Omega(n)$.

(b) (2 *marks*) Explain why an upper bound for $T(n)$ is given by the recurrence:

$$T(n) \leq T(k) + T(n - k - 1) + d.$$

(c) (2 *marks*) Use the substitution method to show that $T(n) = \mathcal{O}(n)$.

Question 4. [4 *marks*]

Suppose the nodes of a Red-Black tree are augmented with an additional attribute min with the following property:

For any node x, $x.min$ points to the node with the minimum key in the subtree rooted at x.

State the changes that need to be made to the RB-INSERT algorithm in order to maintain the min attribute at each node.

Question 5. [4 *marks*]

Let T denote a dynamic storage table that can expand or contract when an element is to be inserted or deleted. Suppose that the TABLE-DELETE operation works as follows: If the load factor α of the table goes below $\frac{1}{4}$, then a new table is declared who size is half the current table's size and all elements are copied to the new table. Let α_i denote the load factor after the ith operation and let Φ be the following potential function:

$$\Phi_i = \begin{cases} 2.num_i - size_i & \text{if } \alpha_i \geq \frac{1}{2} \\ size_i/2 - num_i & \text{if } \alpha_i < \frac{1}{2} \end{cases}$$

Show that the amortized cost of a TABLE-DELETE operation is at most 3, i.e., $\hat{c}_i \leq 3$, in the following cases:

(a) (2 *marks*) $\frac{1}{4} < \alpha_{i-1} < \frac{1}{2}$ and $\frac{1}{4} \leq \alpha_i < \frac{1}{2}$ (no contraction of table)

(b) (2 *marks*) $\alpha_{i-1} = \frac{1}{4}$ and $\alpha_i < \frac{1}{2}$ (contraction of table).

Question 6. [8 *marks*]

(a) (3 *marks*) Give pseudocode for the ANY-SEGMENTS-INTERSECT algorithm for determining if a given set of line segments has a pair of intersecting line segments.

(b) (3 *marks*) Explain why the algorithm runs in $\mathcal{O}(n \log n)$ time.

(c) (2 *marks*) Explain how you could adapt the ANY-SEGMENTS-INTERSECT algorithm to determine if a given set of disks has a pair of intersecting disks. Each disk is defined by a center (x_i, y_i) and a radius r_i.

Question 7. [4 *marks*]

(a) (1 *mark*) Describe the Discrete Fourier Transform (DFT) of a polynomial $A(x)$.

(b) (3 *marks*) Give a high-level description of how the Fast Fourier Transform (FFT) computes the DFT of a polynomial $A(x)$, using the polynomials $A^{[0]}(x)$ and $A^{[1]}(x)$.