Question 1. (5 *marks*)

Suppose you are given an array A containing n d-digit numbers, where each digit can take on up to k possible values.

(a) (3 *marks*) Explain in detail how RADIX-SORT would sort the array A and give the running time of RADIX-SORT.

(c) (2 *marks*) What is the running time of RADIX-SORT if it sorts using r columns at a time, where $1 < r \leq d$? Explain your answer.

Question 2. [6 *marks*]

Consider the following pseudocode for RANDOMIZED-SELECT that takes as input an array A of distinct numbers and an integer i with $1 \leq i \leq A.length$ and returns the ith smallest entry in A.

```
RANDOMIZED-SELECT(A, p, r, i)
if p == r
    return A[p]
q = RANDOMIZED-PARTITION(A, p, r)
k = q - p + 1
if i == k
    return A[q]
elseif i < k
    return RANDOMIZED-SELECT(A, p, q - 1, i)
else return RANDOMIZED-SELECT(A, q + 1, r, i - k)
```

(a) (3 *marks*) Using the indicator random variable $X_k = \mathrm{I}\{A[p, \ldots, q] \text{ has exactly } k \text{ elements}\}$, explain in detail why the following recurrence describes the running time of RANDOMIZED-SELECT.

$$T(n) \leq \sum_{k=1}^{n} X_k \cdot (T(\max(k-1, n-k)) + O(n)).$$

(b) (3 *marks*) Given the following recurrence for the expected running time of RANDOMIZED-SELECT, use the substitution method to show that $\mathrm{E}[T(n)] = \mathcal{O}(n)$:

$$\mathrm{E}[T(n)] \leq \frac{2}{n} \sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} \mathrm{E}[T(k)] + O(n).$$

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*)

Let T be a red-black tree whose nodes are augmented with an attribute f such that for any node x, the value of $x.f$ depends only on the information in x, $x.left$ and $x.right$, possibly including $x.left.f$ and $x.right.f$. Explain why we can maintain the values of f in all nodes of T during insertion and deletion in $\mathcal{O}(\log n)$ time, where n is the number of nodes of T.

Question 5. [5 *marks*]

Let T denote a dynamic storage table that can expand or contract when an element is to be inserted or deleted. TABLE-INSERT works as follows: If that table is full, then a new table is declared who size is double the current table's size and all elements are copied to the new table and the new element is inserted. TABLE-DELETE 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, except the one to be deleted.

Discuss in detail how amortized analysis can be used to show that the cost of n TABLE-INSERT and/or TABLE-DELETE operations is $\mathcal{O}(n)$.
(You do not need to do the $\hat{c}$ calculations, but state what calculations are required.)

Question 6. [5 *marks*]

(a) (3 *marks*) Describe in detail the steps of Jarvis's march algorithm that takes as input a set Q of points in the plane and returns the convex hull of the Q.

(b) (2 *marks*) Give the running-time of Jarvis's march and explain why that is the running-time.

Question 7. (4 *marks*)

(a) (2 *marks*) Describe the Discrete Fourier Transform (DFT) of a polynomial.

(b) (2 *marks*) Give a high-level description of how the Fast Fourier Transform (FFT) algorithm obtains the DFT of a polynomial.