Lab Assignment 13

Please follow the following instructions for coding this assignment. We have multiple functions. Use the following function declarations and return types.

1. `def sequential_search(arr, target)`

- **Input**:
  - `arr` (list): List of elements to search in.
  - `target` (any): The element to find.
- **Output**:
  - `int`: Index of `target` if found; otherwise, `-1`.
- **Description**:
  Performs a linear search through the list to find the `target`.

---

2. `def binary_search(arr, target)`

- **Input**:
  - `arr` (list): A **sorted** list of elements.
  - `target` (any): The element to find.
- **Output**:
  - `int`: Index of `target` if found; otherwise, `-1`.
- **Description**:
  Uses binary search to efficiently locate the `target` in a sorted list.

---

3. `def bubble_sort(arr)`

- **Input**:
  - `arr` (list): List of elements to be sorted.
- **Output**:
  - `list`: The same list, sorted in-place.
- **Description**:
  Sorts the list using bubble sort by repeatedly swapping adjacent elements if they are in the wrong order.

---

4. `def insertion_sort(arr)`

- **Input**:
  - `arr` (list): List of elements to be sorted.
- **Output**:
  - `list`: The same list, sorted in-place.
- **Description**:
  Sorts the list by building a sorted portion one element at a time using insertion.

---

5. `def selection_sort(arr)`

- **Input**:
  - `arr` (list): List of elements to be sorted.
- **Output**:
  - `list`: The same list, sorted in-place.
- **Description**:
  Sorts the list by repeatedly finding the minimum element and moving it to the front.

---

6. `def merge_sort(arr)`

- **Input**:
  - `arr` (list): List of elements to be sorted.
- **Output**:
  - `list`: A new sorted list (in-place modifications happen as well).
- **Description**:
  Sorts the list using the divide-and-conquer approach of merge sort. Splits the list recursively and merges sorted halves.

---

7. `def quick_sort(arr)`

- **Input**:
  - `arr` (list): List of elements to be sorted.
- **Output**:
  - `list`: A new list that is sorted (not in-place).
- **Description**:
  Uses the quick sort algorithm by choosing a pivot, partitioning the list, and recursively sorting the sublists.