Class Test 2 (Memo)

Perform the following sorting algorithms show all steps:

1. Selection sort:
[42, 18, 27, 5, 33] (2)

- Pass 1:
  - Find the minimum value in the list [42, 18, 27, 5, 33], which is 5.
  - Swap 5 with the first element (42).
  - List: [5, 18, 27, 42, 33]
- Pass 2:
  - Find the minimum in the remaining list [18, 27, 42, 33], which is 18.
  - No swap is needed.
  - List: [5, 18, 27, 42, 33]
- Pass 3:
  - Find the minimum in the remaining list [27, 42, 33], which is 27.
  - No swap is needed.
  - List: [5, 18, 27, 42, 33]
- Pass 4:
  - Find the minimum in the remaining list [42, 33], which is 33.
  - Swap 33 with 42.
  - List: [5, 18, 27, 33, 42]
  - Final Sorted List: [5, 18, 27, 33, 42]

2. Bubble Sort:
[8, 15, 3, 21, 7] (2)

- Pass 1:
  - Compare 8 and 15 → Swap. List: [15, 8, 3, 21, 7]
  - Compare 8 and 3 → No swap.
  - Compare 3 and 21 → Swap. List: [15, 8, 21, 3, 7]
  - Compare 3 and 7 → Swap. List: [15, 8, 21, 7, 3]
- Pass 2:
  - Compare 15 and 8 → No swap.
  - Compare 8 and 21 → Swap. List: [15, 21, 8, 7, 3]
  - Compare 8 and 7 → No swap.
  - Compare 7 and 3 → No swap.
- Pass 3:
  - Compare 15 and 21 → Swap. List: [21, 15, 8, 7, 3]
  - Compare 15 and 8 → No swap.
  - Compare 8 and 7 → No swap.
- Pass 4:
  - Compare 21 and 15 → No swap.
  - Compare 15 and 8 → No swap.

Final Sorted List: [21, 15, 8, 7, 3]

3. Insertion Sort:
[9, 4, 6, 2, 10] (2)

- Pass 1: (4 compared with 9)
    - Insert 4 before 9. List: [4, 9, 6, 2, 10]
- Pass 2: (6 compared with 9 and 4)
    - Shift 9 to the right, insert 6 after 4. List: [4, 6, 9, 2, 10]
- Pass 3: (2 compared with 9, 6, and 4)
    - Shift 9 to the right, then 6, then 4. Insert 2 at the beginning. List: [2, 4, 6, 9, 10]
- Pass 4: (10 compared with 9)
    - No insertion needed. List: [2, 4, 6, 9, 10]

Final Sorted List: [2, 4, 6, 9, 10]

4. Merge Sort:
[11, 3, 8, 16, 2, 9, 5, 14] (2)

- Divide:
    - Split into Left: [11, 3, 8, 16], Right: [2, 9, 5, 14]
- Sort Left Half:
    - Split [11, 3, 8, 16] → [11, 3] and [8, 16]
    - Split [11, 3] → [11] and [3] → Merge: [3, 11]
    - Split [8, 16] → [8] and [16] → Merge: [8, 16]
    - Merge [3, 11] and [8, 16] → [3, 8, 11, 16]
- Sort Right Half:
    - Split [2, 9, 5, 14] → [2, 9] and [5, 14]
    - Split [2, 9] → [2] and [9] → Merge: [2, 9]
    - Split [5, 14] → [5] and [14] → Merge: [5, 14]
    - Merge [2, 9] and [5, 14] → [2, 5, 9, 14]
- Merge All:
    - Merge [3, 8, 11, 16] and [2, 5, 9, 14] → [2, 3, 5, 8, 9, 11, 14, 16]

Final Sorted List: [2, 3, 5, 8, 9, 11, 14, 16]

5. Explain one advantage of using Merge Sort over Bubble Sort when sorting a large dataset ? (2)

Merge Sort is more efficient for large datasets because it uses the “divide and conquer” strategy, which breaks the list down into smaller sublists, sorts them, and then merges them back together.