**Beginner Level:**

1. Implement Bubble Sort
2. Implement Selection Sort
3. Implement Insertion Sort
4. Implement Merge Sort
5. Implement Quick Sort
6. Implement Heap Sort
7. Implement Counting Sort
8. Implement Radix Sort
9. Implement Bucket Sort
10. Implement Shell Sort
11. Sort an array of 0s, 1s, and 2s (Dutch National Flag problem)
12. Find the kth smallest/largest element using Quickselect
13. Sort a nearly sorted (or K sorted) array
14. Sort an array of strings
15. Sort an array of dates
16. Sort an array using a custom comparator
17. Implement Pancake Sort
18. Sort an array by frequency of elements
19. Sort an array of objects by a key
20. Sort an array of integers by the number of 1s in their binary representation

**Medium Level:**

1. Implement 3-way Quick Sort for arrays with many duplicate keys
2. Sort a linked list using Merge Sort
3. Sort a linked list using Quick Sort
4. Sort a stack using recursion
5. Implement Tim Sort
6. Implement IntroSort (Introspective Sort)
7. Sort an array of large numbers (where each number is represented as a string)
8. Sort an array of points by their distance from the origin
9. Sort an array of fractions
10. Implement Bitonic Sort
11. Implement Cocktail Shaker Sort
12. Implement Comb Sort
13. Implement Gnome Sort
14. Implement Pigeonhole Sort
15. Implement Stooge Sort
16. Implement Tree Sort using Binary Search Tree
17. Implement BogoSort
18. Implement Cycle Sort
19. Implement Flash Sort
20. Implement Patience Sort

**Advanced Level:**

1. Implement Strand Sort
2. Implement Bead Sort (Gravity Sort)
3. Implement Library Sort
4. Implement Sleep Sort
5. Implement Spaghetti Sort
6. Implement Burst Sort
7. Implement Spread Sort
8. Implement Flashsort
9. Implement Smoothsort
10. Implement Parallel Sorting using multi-threading