1. Huffman coding
2. 111001100
3. EFG
4. Mergesort
5. O(nlogn)
6. O(nlogn)
7. O(nlogn)
8. Shellsort
9. **7-sort:** 2, 1, 7, 6, 5, 4, 3, 9, 8
10. **3-sort:** 2, 1, 4, 3, 5, 7, 6, 9, 8
11. **1-sort:** 1, 2, 3, 4, 5, 6, 7, 8, 9
12. Insertion sort
13. 3, 1, 4, 1, 5, 9, 2, 6, 5

1, 3, 4, 1, 5, 9, 2, 6, 5

1, 3, 1, 4, 5, 9, 2, 6, 5

1, 1, 3, 4, 5, 9, 2, 6, 5

1, 1, 3, 4, 5, 2, 9, 6, 5

1, 1, 3, 4, 2, 5, 9, 6, 5

1, 1, 3, 2, 4, 5, 9, 6, 5

1, 1, 2, 3, 4, 5, 9, 6, 5

1, 1, 2, 3, 4, 5, 6, 9, 5

1, 1, 2, 3, 4, 5, 6, 5, 9

1, 1, 2, 3, 4, 5, 5, 6, 9

1. O(n) because it will compare the values next to each other one time, and one time only, and it complete that comparison operation n times for all the elements
2. Quicksort
3. novakc