1. Binary search fails without order because the mid value needs to tell you direction. With no order, it is random guessing.

2. A hash table uses chains or probing to keep keys from piling up. This stops slow lookup lines.

3. A min-heap maintains order by up-heap and down-heap moves. Those swaps keep the root the smallest.

4. BFS guarantees the least number of edges because of its level expansion. Every new level is one more step.

5. Quicksort is fast when splits are even. It is slow when data is already sorted or badly arranged.

6. Linked lists insert fast by pointer updates. Arrays copy many elements and cost more.

7. DP works because saved answers get reused. The more overlap, the better the gain.

8. DFS stacks nodes so it explores deep paths before others. The structure sets the travel pattern.

9. A trie stores characters in paths. Shared prefixes save time in lookups.

10. Balanced BSTs avoid long lines. Unbalanced ones behave like lists.