1. Binary search only works when order is known. Without order, you cannot discard half.

2. A hash table keeps collisions small with chains or probes. This matters to keep lookup time steady.

3. The min-heap uses swaps to fix broken order. Insert and delete both do this.

4. BFS finds shortest edge paths because of level scanning. Each level adds one more hop.

5. Quicksort is fast with semi-random data. Worst case appears in sorted or patterned data.

6. Linked lists can insert fast in the middle. Arrays cannot because they shift things.

7. DP stores subproblem results. Overlap means less wasted work.

8. DFS uses a stack to pick the next node. It goes deep and returns later.

9. A trie stores strings by letter nodes. This makes prefix search simple.

10. Balanced BSTs keep height low. If they skew, speed drops.