1. Without sort, the mid step cannot decide where to go. Sorting gives meaning to half splits.

2. Collisions are fixed with chains or probes. Speed drops when too many keys fight for slots.

3. The heap floats new items up and pushes removed items down. This keeps the top small.

4. BFS moves outward one ring at a time. That ring count is the path length.

5. Quicksort wins when splits are even, fails when splits are near 0 and n.

6. Linked lists insert by changing next pointers. Arrays copy blocks.

7. DP reuses old results. Overlap is what makes this useful.

8. DFS uses a stack to go deep before wide. That is why it finds long paths first.

9. Tries store each letter in a level chain. This helps autocomplete tasks.

10. Balanced BSTs stop long searches. Unbalanced trees become slow.