1. Binary search only works when you know which half the target might be in. Without sorting, cutting the range gives no real info.

2. Collisions are handled with lists or jumps. Without a plan for them, lookups can turn into long searches.

3. Inserts bubble up, deletes bubble down. This keeps the smallest item always at the root.

4. BFS reaches nodes in rising distance. This makes the first visit to a node the shortest path.

5. Good pivots give fast splits, bad pivots give long chains. That is why the worst case is slow.

6. In a linked list you insert by linking a new node between two others. Arrays must shift elements.

7. DP reuses results. It works when small parts are solved many times.

8. DFS uses a stack so it dives deep before backtracking. The stack tracks where to return.

9. A trie has each level store one character. Prefixes share space.

10. Balanced trees keep height low. Unbalanced trees grow tall and slow.