Binary search breaks on unsorted data because the mid point has no meaning. Sorting gives structure.

Collisions need chaining or probes so the bucket does not overflow. This protects constant time lookups.

Insert bubbles up, delete bubbles down. That keeps min-heap order correct.

BFS explores neighbors layer by layer. It gives shortest edge paths for this reason.

Quicksort works when each pivot splits well. If every pivot is bad, it becomes slow.

Linked lists insert with pointer moves. Arrays need to shift blocks.

DP uses past work to skip calls. Overlap means many parts repeat.

DFS uses a stack to dive deep. The stack sets the order in which nodes show up.

A trie stores letters along branches. Prefix search is quick because the path is shared.

Balanced BSTs keep height short. If they lean, operations degrade.