Binary search needs sorted order so the mid check tells direction. If not sorted, you cannot cut the search down.

Hash tables handle collisions by chaining or probing. Without this, buckets fill and lookups slow.

A min-heap swaps nodes until parent <= child. This keeps the root as the min.

BFS uses levels to track distance. First visit is shortest.

Quicksort is fast with random pivots. Worst case is bad pivot every time.

Linked lists insert by linking nodes. Arrays must push elements over.

DP caches answers. Many repeats make it worth it.

DFS uses a stack to track its deep path. This changes the route taken.

A trie stores strings through character edges. Prefix work speeds up.

Balanced BSTs keep log height. Unbalanced becomes long.