

Binary search needs order so it knows where to go next. If not sorted, it loses all logic.

A hash table uses chains or probes to control collisions. This keeps lookup time small.

The heap fixes itself by swapping nodes with parents or children. This keeps the min at the top.

BFS works level by level, so each layer is one more edge away. That gives shortest paths.

Quicksort is great when pivot picks are good. Very bad pivots force near linear splits.

Linked lists insert easy because pointers can change fast. Arrays must slide many items.

DP solves each subproblem once. Overlap means you save many calls.

DFS uses a stack to push new deep nodes. This makes it dive before exploring sideways.

Tries keep words in shared prefix branches. This makes prefix tasks simple.

Balanced BSTs stop long chains. Unbalanced trees waste time.