

Sorting ensures each step cuts the answer space. Without it you cannot throw away half the array.

Collisions are solved by linking items or trying new slots. They matter because too many slow the whole table.

A min-heap swaps items up or down. This stops larger items from sitting above smaller ones.

BFS finds routes in layers. The first time it hits a node is the shortest distance.

Quicksort depends on good pivot splits. Worst case splits ruin the run time.

Linked lists let you insert in the middle by reconnecting nodes. Arrays need big shifts.

DP uses stored values from past work. It shines where repeats are common.

DFS needs a stack to go deep then backtrack. This affects which nodes get seen first.

Tries break words into character steps. Prefix search is fast because of shared paths.

Balanced trees keep search paths short. If they tilt, the height becomes large.