

Algorithm patterns cheat sheet

Read the problem and find its words in the second column. That row is your pattern. The last column names the pattern it gets mixed up with, so check that one before you write code.

One row per pattern, grouped by topic, 52 in all. Each name opens the full page with a template in six languages.

Pattern	Words in the problem	Time	Not to confuse with
ARRAYS			
In-place array transform	modify the array in place · $O(1)$ extra space · do not allocate another array/string · reverse/rotate/partition/move elements over a single buffer · return the new length after removing duplicates	$O(n)$	Two pointers (same direction)
HASHING			
Hash set / map	have I seen this before · count how many of each · find duplicates · does a matching/complement value exist · data is in any order / unsorted	$O(n)$	Two pointers (opposite ends)
TWO POINTERS			
Two pointers (opposite ends)	sorted array (or you are allowed to sort) · find a pair/triple with a target sum · squeeze from both ends (container, area) · palindrome check	$O(n)$	Sliding window (variable)
Two pointers (same direction)	remove duplicates in place, keep order · move/partition elements while preserving order · compact an array and use $O(1)$ extra space · shift zeros or a target value to the end	$O(n)$	In-place array transform
SLIDING WINDOW			
Sliding window (fixed size)	contiguous subarray/substring of a given length k · every window of size k · rolling average or moving sum · min/max/count/sum over each fixed span	$O(n)$	Sliding window (variable)
Sliding window (variable)	longest/shortest contiguous subarray or substring · satisfying a condition (sum $\leq K$, at most, without breaking a limit) · at most K distinct, no repeated element, contains all of X · contiguous span, not a subsequence	$O(n)$	Sliding window (fixed size)
PREFIX SUMS			
Prefix sums	many sum-over-a-range queries · answer several range-sum questions on the same array · sum from index l to r · running total or cumulative sum	$O(n)$ build / $O(1)$ query	Sliding window (fixed size)
Difference array	many range updates first, then read the final array once · add a value to every element in a range, repeated many times · apply k range increments before answering any query · booking/interval counters over an array	$O(n)$	Prefix sums
SEARCHING			
Linear search	unsorted array · check every element · no order guarantee · find first/any match · small n or one-time scan	$O(n)$	Binary search (array)
Binary search (array)	sorted array · find a target / first or last occurrence · n up to $1e6+$ and $O(\log n)$ expected · find the boundary where a condition flips · search a rotated-but-mostly-sorted array (with a twist)	$O(\log n)$	Linear search
Binary search on a rotated array	sorted array rotated at an unknown pivot · find a target in $O(\log n)$ · no duplicates (or handle them separately) · find the minimum / the rotation point · array increases then drops once	$O(\log n)$	Binary search (array)
Binary search on the answer	minimize the maximum / maximize the minimum · smallest value that satisfies a condition · find the smallest capacity/speed/days that works · answer lies in a numeric range, not an array position · a feasibility check is monotonic (true after some point)	$O(n \log(\text{range}))$	Binary search (array)

Pattern	Words in the problem	Time	Not to confuse with
SORTING			
Elementary sorts (selection, bubble, insertion)	n is small (tens or a few hundred) · array is nearly sorted already · sort in place with O(1) extra space · teaching/interview question about how sorting works · stability matters and simplicity is fine	$O(n^2)$	Fast sort (merge / quick)
Fast sort (merge / quick)	sort an array of real size (thousands+) · need $O(n \log n)$ guaranteed or expected · sort as a subroutine before another algorithm (two pointers, greedy, sweep) · stability required (merge sort) or in-place expected (quicksort) · just "sort this" with no small-n or special-key hint	$O(n \log n)$	Elementary sorts (selection, bubble, insertion)
Non-comparison sort (counting / radix)	keys are integers in a small known range · sort ages, grades, or bounded counts · need $O(n)$ and comparisons are the bottleneck · sort by digit or fixed-width key · n large but the value range is much smaller than n	$O(n)$	Fast sort (merge / quick)
Sort with a custom comparator	sort by multiple fields / tie-break on a second key · custom order (not plain ascending) · sort descending, or by a computed key · "sort so that X comes before Y when..." · reorder objects/records, not raw numbers	$O(n \log n)$	Fast sort (merge / quick)
INTERVALS			
Intervals: merge & insert	merge overlapping intervals · insert a new interval into a sorted list · meetings, bookings, or ranges that overlap · given as [start, end] pairs · free time / busy time between intervals	$O(n \log n)$	Sweep line (event counting)
Sweep line (event counting)	maximum number of overlapping meetings/intervals at once · minimum rooms/resources needed · busiest moment in time · how many intervals cover a given point · events happening at the same time	$O(n \log n)$	Intervals: merge & insert
LINKED LISTS			
Linked list reversal	reverse a linked list (whole list or between positions) · reverse in groups of k · no extra array or O(1) extra space allowed · swap pairs of nodes · singly linked list, only .next available	$O(n)$	Linked list merge & reorder
Fast & slow pointers	detect a cycle in a linked list · find where a cycle begins · find the middle node of a list · determine if a list is a palindrome · no extra memory / O(1) space, only pointers	$O(n)$	Two pointers (same direction)
Linked list merge & reorder	merge two sorted linked lists · merge k sorted lists · reorder a list (interleave front and back halves) · rearrange nodes without copying values into an array · dummy head / sentinel node	$O(n + m)$	Linked list reversal
STACKS AND QUEUES			
Stack (LIFO)	valid parentheses / balanced brackets · match the most recent open with the next close · undo, backtrack to the last state, nested structure · evaluate an expression with nested operators · process in reverse order of arrival	$O(n)$	Monotonic stack
Monotonic stack	next greater / next smaller element · daily temperatures until it gets warmer · largest rectangle in a histogram · trap rainwater between bars · span/streak that ends at the first bigger or smaller value	$O(n)$	Stack
Monotonic deque (sliding window max/min)	maximum or minimum of every window of size k · sliding window maximum/minimum · shortest subarray with a sum at least K · need both ends of a queue popped/pushed · deque, double-ended queue	$O(n)$	Monotonic stack
RECURSION			
Recursion	problem is defined in terms of a smaller version of itself · nested/tree-shaped structure to walk (directory, expression, linked list) · compute a count/sum/depth over all sub-elements · natural base case + recursive case split (factorial, Fibonacci, tree depth)	$O(n)$	Recursion with memoization
Recursion with memoization	recursive calls repeat the same arguments (overlapping subproblems) · plain recursion is correct but too slow / exponential blow-up · small number of distinct states even though the recursion tree is huge · cache/memo the result of (i, ...) before returning · Fibonacci / climbing stairs / grid-path wording with a hint to cache	$O(\text{states})$	Recursion
BACKTRACKING			
Backtracking	generate all subsets/permutations/combinations · n is small (roughly $n \leq 12-20$), exponential search is acceptable · make a choice, recurse, then undo it before trying the next choice · place items under constraints and back off on conflict (N-Queens, Sudoku) · return every valid arrangement, not just one	$O(2^n)/O(n!)$	Recursion

Pattern	Words in the problem	Time	Not to confuse with
TREES			
Tree traversal	binary tree given as a root node with left/right pointers · visit every node (preorder / inorder / postorder / level-order) · compute a depth, height, or sum over the whole tree · process children before or after the parent · level-by-level output (BFS with a queue)	$O(n)$	Graph BFS / DFS
Binary search tree	binary search tree: left < node < right at every node · insert/delete/search while keeping the data ordered · validate BST / kth smallest / inorder traversal gives sorted order · closest value or range query on a set that changes over time · no array given up front, values arrive one at a time	$O(\log n)$ avg	Binary search (array)
Lowest common ancestor	lowest common ancestor of two nodes · deepest node that is an ancestor of both p and q · each recursive call returns which target(s) it found below it · path from root to each node then compare (alternate approach) · binary tree or BST given, find where the two paths split	$O(n)$	Binary search tree
Tree serialize / deserialize	serialize a tree to a string and deserialize it back · rebuild the exact same shape, not just the same values · encode/decode wording, design a codec for a tree · null/None markers for missing children · store or transmit a tree (network transfer, file, cache)	$O(n)$	Tree traversal
HEAPS AND TOP-K			
Binary heap / priority queue	always need the current smallest/largest · top k elements · merge k sorted lists · running median while items keep arriving · priority queue / most urgent next	$O(\log n)$ per op	Fast sort (merge / quick)
GRAPHS			
Graph BFS / DFS	shortest path in an unweighted graph · fewest steps/hops between nodes · can node X reach node Y · explore ring by ring / layer by layer · graph given as adjacency list, may contain cycles	$O(V+E)$	Tree traversal
Connected components	count separate groups/islands/clusters · graph may be disconnected · how many connected pieces · friend circles · label connected regions	$O(V+E)$	Graph BFS / DFS
Topological sort (Kahn's algorithm)	task must come before another · order that respects dependencies · detect a cycle in prerequisites · build order / course schedule / install order · directed graph, X must happen before Y	$O(V+E)$	Graph BFS / DFS
Union-find (disjoint set)	are these two in the same group · merge/union facts arrive one at a time · dynamic connectivity as edges are added · account/friend merging over time · adding this edge would create a cycle	near $O(1)$ per op	Connected components
Dijkstra's algorithm	cheapest/shortest path with weighted edges · non-negative edge weights · minimum total cost to reach every node · network delay time / route cost, not fewest roads · weighted graph, single-source shortest paths	$O(E \log V)$	Graph BFS / DFS
Eulerian path (Hierholzer's algorithm)	use every edge exactly once · reconstruct an itinerary from tickets · one-stroke drawing / draw without lifting the pen · walk every road/domino exactly once · flights or dominoes must chain into one trip	$O(E)$	Graph BFS / DFS
Bipartite check (two-coloring)	split into two groups with no conflict inside a group · two-color the graph · detect an odd-length cycle · us vs them / two-shift scheduling / two-room assignment · possible bipartition of conflict pairs	$O(V+E)$	Graph BFS / DFS
GREEDY			
Greedy (exchange argument)	fewest coins or pieces to make an amount · largest that still fits, taken greedily · make change with denominations · prove optimality with an exchange argument	$O(n \log n)$	Dynamic programming (1-D)
DYNAMIC PROGRAMMING			
Dynamic programming (1-D)	best/maximum/minimum ending at position i · ways to reach, climb, or tile · cannot pick two adjacent items · minimum cost path through a grid · overlapping subproblems, same state recomputed	$O(\text{states})$	Recursion with memoization
DP on subsequences (LIS / LCS / edit distance)	longest increasing subsequence · longest common subsequence between two strings · minimum edits to turn one string into another · keep relative order, may skip items · diff or alignment between two sequences	$O(n^2)$ / $O(nm)$	Dynamic programming (1-D)
DP on intervals	combine adjacent pieces at some cost · best way to split a range into two parts · matrix chain multiplication order · merge cost depends on the whole span · optimal parenthesization or build order	$O(n^3)$	Dynamic programming (1-D)
DP on trees	best choice over a hierarchy or org chart · cannot pick a node and its direct parent together · post-order: children settled before the parent · each node combines states from its children	$O(n)$	Tree traversal

Pattern	Words in the problem	Time	Not to confuse with
STRINGS AND DYNAMIC PROGRAMMING			
DP on strings (word break)	can the string be split into dictionary words · segment a string using a word list · boolean reachable at each cut position · glued text needs word boundaries	$O(n^2)$	DP on subsequences (LIS / LCS / edit distance)
STRINGS			
Trie (prefix tree)	autocomplete or typeahead over many strings · longest common prefix among words · insert, search, and starts-with on a word set · match against a large fixed dictionary by prefix	$O(\text{len})$ per op	Hash set / map
BIT MANIPULATION			
Bit manipulation	find the single or unique number using XOR · set, clear, toggle, or test a bit · pack many yes/no flags into one integer · count set bits or check a power of two	$O(1)/O(n)$	Hash set / map
MATH AND NUMBER THEORY			
Math and number theory (GCD, sieve, modular)	reduce a fraction to lowest terms · list all primes up to n · answer modulo $1e9+7$ · when do two repeating intervals line up again (LCM)	varies	Bit manipulation
MATRIX			
Matrix traversal (spiral / diagonal)	read or fill a grid in spiral order · walk a matrix diagonal by diagonal · rotate a matrix 90 degrees · visit every cell in a fixed geometric order	$O(mn)$	Matrix search (sorted grid)
Matrix search (sorted grid)	matrix sorted along both rows and columns · search for a value in a 2D sorted grid · start from a corner and eliminate a row or column · row and column both monotonic	$O(m+n)$	Binary search (array)
Matrix word search (grid DFS/backtracking)	does the word exist as a path on the board · trace letters through adjacent cells, no reuse · DFS with backtracking on a grid · mark a cell as visited, then undo it	$O(mn \cdot 4^L)$	Backtracking
ADVANCED			
Fenwick tree / segment tree	range sum or query with point updates interleaved · live leaderboard or running rank while scores change · count inversions or how many earlier values are larger · range queries while the data keeps changing	$O(\log n)$ per op	Prefix sums