

Interview revision sheet

The topics that come up most in interviews, on one page. These are typical costs, not guarantees: the reference pages list the assumptions.

Data structures

Topic	Time	Space
Dynamic array	Index $O(1)$; append $O(1)$ amortized; insert or delete in the middle $O(n)$	$O(n)$
Hash table (dict and set)	Lookup, insert, delete $O(1)$ expected; $O(n)$ worst case	$O(n)$
Stack	Push, pop, peek $O(1)$	$O(n)$
Queue and deque	Enqueue and dequeue $O(1)$ with a deque	$O(n)$
Binary heap and priority queue	Peek $O(1)$; push and pop $O(\log n)$; build from n items $O(n)$	$O(n)$
Binary search tree	Search, insert, delete $O(h)$: $O(\log n)$ balanced, $O(n)$ worst case	$O(n)$
Union-Find (disjoint-set union)	Amortized $O(\alpha(n))$ per find or union	$O(n)$
Monotonic stack	$O(n)$ total	$O(n)$

Sorting

Topic	Time	Space
Mergesort	$\Theta(n \log n)$ in every case	$O(n)$

Searching

Topic	Time	Space
Binary search	$O(\log n)$	$O(1)$ iterative

Graphs and trees

Topic	Time	Space
Breadth-first search (BFS)	$O(V + E)$	$O(V)$
Depth-first search (DFS)	$O(V + E)$	$O(V)$
Tree traversals	$O(n)$	$O(h)$ for depth-first orders
Dijkstra's algorithm	$O((V + E) \log V)$ with a binary heap	$O(V + E)$
Graph representations	Edge check: matrix $O(1)$, list $O(\deg(u))$	List $O(V + E)$; matrix $O(V^2)$

Patterns

Topic	Time	Space
Two pointers	$O(n)$	$O(1)$
Sliding window	$O(n)$	$O(1)$ or $O(n)$
Prefix sums	$O(n)$ to build; $O(1)$ per range sum	$O(n)$
Hash map frequency counting	Expected $O(n)$ total	$O(n)$
Backtracking and subsets	$O(2^n)$ subsets; $O(n \cdot 2^n)$ including copying each one	$O(n)$ recursion, plus stored output
Dynamic programming	Number of states $\times$ transitions per state	Depends on memoization or tabulation
Merge intervals	$O(n \log n)$	$O(n)$

Rough guide: input size to complexity

Input size n	Complexity that usually fits
Up to about 10	$O(n!)$
Up to about 20	$O(2^n)$
Up to about 500	$O(n^3)$
Up to about 5,000	$O(n^2)$
Up to about 1,000,000	$O(n \log n)$
Up to about 100,000,000	$O(n)$

Assumes roughly 10^8 simple operations per second. Real limits depend on the language and the work per step.