

Category	QueryType	Example	Common Data Structures / Techniques	Typical TimeCompl
RangeAggregation	Range Sum	$\text{sum}(l,r)$	Prefix Sum, Fenwick Tree, Segment Tree	$O(1) / O(\log n)$
RangeAggregation	Range Minimum	$\text{min}(l,r)$	Segment Tree, Sparse Table	$O(1)$ (static) / $O(\log n)$
RangeAggregation	Range Maximum	$\text{max}(l,r)$	Segment Tree, Sparse Table	$O(1)$ (static) / $O(\log n)$
RangeAggregation	Range GCD	$\text{gcd}(l,r)$	Segment Tree, Sparse Table	$O(1) / O(\log n)$
RangeAggregation	Range LCM	$\text{lcm}(l,r)$	Segment Tree	$O(\log n)$
RangeAggregation	Range XOR	$\text{xor}(l,r)$	Prefix XOR, Segment Tree	$O(1) / O(\log n)$
RangeAggregation	Range AND/OR	$\text{and}(l,r)$, $\text{or}(l,r)$	Segment Tree, Prefix OR	$O(\log n)$
RangeAggregation	Range Product with mod	$\text{prod}(l,r) \% \text{mod}$	Segment Tree	$O(\log n)$
RangeAggregation	Range Median	$\text{median}(l,r)$	Wavelet Tree, Merge Sort Tree, Order-static tree	$O(\log n) - O(\log^2 n)$
RangeAggregation	Range Variance	$\text{var}(l,r)$	Prefix moments, Segment Tree	$O(1) / O(\log n)$
RangeAggregation	Range DotProduct	$a[i]*b[i]$ over $[l,r]$	SegTree of pairs, FFT for blocks	$O(\log n) / \text{batch FFT}$
RangeFrequency	Count above/below compare k	$\text{count}(a[i]>k, l, r)$	Merge Sort Tree, BIT, order-stat set	$O(\log^2 n)$
RangeFrequency	Frequency of a value	$\text{freq}(x,l,r)$	Mo's, Wavelet Tree, Offline hashing	$O((n+q)\sqrt{n}) / O(\log n)$
RangeFrequency	Number of distinct value	$\text{distinct}(l,r)$	Mo's, Fenwick + last-pos	$O((n+q)\sqrt{n}) / O(\log n)$
RangeUpdate	Add to Range	add v to $[l,r]$	Lazy Segment Tree, Range BIT	$O(\log n)$
RangeUpdate	Set Range	set $[l,r]=v$	Lazy Segment Tree (assign)	$O(\log n)$
RangeUpdate	Multiply Range	mul v to $[l,r]$	Lazy Segment Tree (mul)	$O(\log n)$
RangeUpdate	Online Transform	$a[i] = a*a + b$ on $[l,r]$	SegTree with matrix-lazy	$O(\log n)$
RangeUpdate	Modulo Range with v	$a[i] \% = v$ on $[l,r]$	Segment Tree with early-break	amortized $O(\log n)$
RangeUpdate	Bitwise Range Operations	XOR/AND/OR to $[l,r]$	Lazy Segment Tree (bitwise)	$O(\log n)$
RangeUpdate	Reverse/Rotate Range	reverse/rotate subarray	Implicit Treap, Splay, Rope	$O(\log n)$
RangeUpdate	Range Insert/Delete	insert pos / delete pos	Implicit Treap, Rope	$O(\log n)$ per op
Order/Position	K-th smallest/largest	$\text{kth}(l,r)$	Merge Sort Tree, Wavelet, PBDS, Segment Tree	$O(\log^2 n) / O(\log n)$
Order/Position	Rank of value	$\text{rank}(x,l,r)$	Order-stat trees, Wavelet tree	$O(\log n)$
Order/Position	Predecessor/Successor	largest $< x$ in $[l,r]$	Balanced BST, Merge Sort Tree	$O(\log n)$
Order/Position	Count pairs/triples	pairs with sum x k in $[l,r]$	Offline divide & conquer, BIT	$O(n \log n)$ batch
Prefix/Suffix	Prefix-sums/moments	$\text{sum}(1,r)$, $\text{sum}(i^2)$	Prefix arrays, precomputed formulas	$O(1)$
Prefix/Suffix	Longest prefix with property	longest prefix sum $\geq k$	Segment Tree + binary search	$O(\log n)$
StringQueries	Substring Hash / Compare	compare $s[l..r] == t$	Rolling Hash, Suffix Array, Segment Tree	$O(1) / O(\log n)$
String/Trie	Trie (prefix lookup, insert, autocomplete)	insert("apple"), startsWith("app")	Array/map-based Trie, compressed children	$O(s)$

Category	QueryType	Example	Common Data Structures / Techniques	Typical TimeCompl
String/Trie	Compressed Trie / Radix Tree	store many keys compactly	Patricia Trie / Radix Tree	$O(s)$ (smaller constants)
String/Trie	Suffix Trie	build all suffixes	Trie of suffixes (rare)	Query $O(pattern)$, memory $O(n^2)$ (impractical)
String/Trie	Suffix Tree (Ukkonen)	find substring, LCS	Suffix Tree + suffix links	Build $O(n)$, query $O(pattern)$
String/Trie	Suffix Array + LCP	substring search, kth suffix	SA + LCP + RMQ	Build $O(n \log n)/O(n)$, query $O(pattern \log n)$ or $O(pattern + \log n)$
String/Trie	Suffix Automaton (SAM)	count distinct substrings, occurrences	SAM (end-pos automaton)	Build $O(n)$, query $O(pattern)$
String/Trie	Aho–Corasick	multi-pattern match in text	Trie + failure links	Build $O(\Sigma P)$, search $O(text + matches)$
String/Trie	DAWG / Minimal DFA	minimal substring automaton	DAWG construction	$O(n)$ build, compact representation
String/Trie	Wavelet Matrix	rank/select over large alphabets, kth	Wavelet Matrix (alt to wavelet tree)	$O(\log \sigma)$ per op
String/Trie	FM-index / BWT	compressed full-text index	BWT + rank/select structures	low-memory substring search, $\sim O(pattern)$
Probabilistic / Streaming	Bloom Filter	membership test (possible FP)	bitset + k hashes	$O(k)$ per query
Probabilistic / Streaming	Counting Bloom / Cuckoo Filter	membership with deletes	Counting bits or cuckoo buckets	$O(1)$ expected
Probabilistic / Streaming	Count–Min Sketch	approximate frequency queries	hashed counter arrays	$O(1)$ per update/query, additive error
Probabilistic / Streaming	HyperLogLog	distinct count (cardinality)	probabilistic registers	$O(1)$ space per register
Approx String Matching	BK-tree (edit distance index)	find words within edit distance k	BK-tree over words	Query depends on k, usually much faster than brute
Approx String Matching	Levenshtein automaton / Myers bit-parallel	small-edit matches	bit-parallel algorithms, automata	Very fast for small k, near $O(s)$ times small factor
Graph / Tree Dynamic	Link-Cut Tree	link(u,v), cut(u,v), path-query	Link-Cut (splay/ETT variants)	Amortized $O(\log n)$
Graph / Tree Dynamic	Euler Tour Tree	dynamic connectivity in forests	Euler Tour + balanced BST	$O(\log n)$ per op
Graph / Tree Dynamic	Heavy-Light Decomposition (HLD)	path queries on trees	HLD + Segment Tree / BIT	$O(\log^2 n)$ or $O(\log n)$ with segtree tweaks
Range/Persistent	Persistent Segment Tree	historical versions, kth in prefix	Persistent segtree (copy-on-write nodes)	$O(\log n)$ per update/query

Category	QueryType	Example	Common Data Structures / Techniques	Typical Time Compl
Range/Persistent	Segment Tree Beats	range chmin/chmax + sum	Advanced segtree invariants	Amortized $\sim O(\log n)$
Spatial / NN	k-d Tree	k-NN and range queries (low-dim)	k-d tree	Avg $O(\log n)$, worst-case higher
Spatial / NN	R-Tree / QuadTree / Octree	spatial indexing / rectangle queries	R-tree / QuadTree / Octree	Practical $\sim O(\log n)$
Spatial / NN	VP-tree / Cover tree / Ball tree	metric nearest neighbor	Metric trees	Good for certain metrics, complexities vary
Misc / Sequence	Rope / Implicit Treap / Treap	mutable string / sequence ops	Implicit Treap, Rope, Splay	$O(\log n)$ per op
Misc / Ordered	PBDS / order-stat tree	order_of_key, find_by_order	GNU PBDS, OST via treap	$O(\log n)$
Misc / Hashing	Cuckoo / Hopscotch Hashing	high load $O(1)$ lookup	Cuckoo tables, hopscotch	$O(1)$ expected
Misc / Others	Skip List	randomized balanced ordered list	Skip list	$O(\log n)$ expected
Graph / Indexing	Centroid Decomposition	path/count queries on tree	Centroid Decomposition + frequency arrays	$O(\log n)$ per update/query typical
Graph / Indexing	LCA (binary lifting / RMQ on Euler tour)	lowest common ancestor queries	Binary lifting, Euler+RMQ	$O(1)$ or $O(\log n)$ per query after preprocessing
Static RMQ	Sparse Table	static range min/max	Sparse Table	$O(1)$ query, $O(n \log n)$ build
Fractional/Cascade	Fractional Cascading	speed multi-level search	Fractional cascading technique	reduces \log^2 to \log in layered searches