Keywords and algorithms

Keywords	Algorithms	Complexity
Sorted array; Search in Binary Search Trees	Binary search Two pointers	O(log(n)) O(n)
Contain duplicates Contains anagrams Intervals merge/insert	Sorting	
(Longest, largest sum) sub-strings, Continuous subsequence/ subarray Interval within an array Linked list for delete, copy, swap, find loops, merge, Palindromes Two linked lists / Two arrays Sorted array (including remove duplicates) Binary search Rotation. Swap i.e.	Two pointers	O(n) <-> O(n^2)

Area computation		
Longest/largest/optimal non-continuous sub-array Linear time complexity Recursive structure with overlapping	Dynamic programming	O(n)
Recursive structure with no overlapping binary trees Quick sort/ Quick rank K-th smallest / largest Find target	Divide-and-conquer	O(n log(n))
(K-th) most frequent Merge with ordering Median Order of Data streams	Heap (min heap or max heap)	O(n*log(K)) or O(n log(n))
least recent	Dynamic queue (list, queue etc)	
Retrieval Back-tracking	Stack	

Non-recursive implémentation Match parentheses		
Binary search tree	Divide-and-conquer, Min and Max of subtrees recursion, In-order traversal, pre-order traversal DFS Min of LST < Max of LST < root < Min of RST < Max of RST	O(n log(n)) Master algorithm T(n) = aT(n/b) + O(f(n)) If f(n) > n^log_b(a) => O(f(n)) If n^log_b(a) > f(n) => O(n^log_b(a)) If n^log_b(a) == f(n) => O(n^log_b(a)* log(n))
Binary tree	Divide-and-conquer Recursion Path Sum DFS/BFS Hash table	
Depth of trees; count subtrees; Path (Sum) of trees /graphs with some properties Root-to-leaf	DFS	O(m*n) where m:= edges, n:= nodes

Find loop in graph; topological ordering Nested structure ([a [b ..]]) Parenthese		
Count spanning trees Count connected sub-regions	Union find	
Shortest path/distance btw pos/ to all building .. Maze Graph with obstacles Minimal distance / Minimal height Level order in Binary tree	BFS	
Permutation Combinations Subsets All possible solutions	Backtracking	Usually in exponential
Longest/most/optimal + sub-string Distance/Match regarding two strings involving sub-strings	Dynamic programming	

Array not sorted and do not want to sort it Return number of uniques paths /ways Return if possible Return longest/largest/ subarray Interval but Cannot decide where to stop/leaves, Interval but Cannot decide where to begin/roots/ Recursive structure with overlapping	Dynamic programming	O(n) in time O(n) in space
Duplicates, not-sorted Anagrams/ invariant to permutations Inverse mapping Track data/ address O(1) implementation Numbers are bounded from [1,2,.., n], n=size of array, to find duplicates, or missing	Hash table Array as hash table	O(n) in time O(n) in space
