

Basics:

Problem	Pattern	Trick	TC	SC
Contains Duplicate	HashSet	if x in seen: return True	O(n)	O(n)
Remove Duplicates Sorted	Two pointers	Overwrite duplicates	O(n)	O(1)
Find Duplicate Number	Floyd's Tortoise / Cycle Detection	Slow + Fast → meet → start → duplicate	O(n)	O(1)

Searching Pairs:

Pivot Index	Prefix sum	leftsum == totalsum-leftsum-n	O(n)	O(1)
Two Sum (unsorted)	HashMap	Store complement	O(n)	O(n)
Two Sum II (sorted)	Two pointers	l,r → move pointers	O(n)	O(1)

Frequency & Majority:

Majority Element	Moore Voting	Candidate + count	O(n)	O(1)
Maximum Subarray	Kadane	cur=max(num,cur+num) → maxAns=max(maxAns,cur)	O(n)	O(1)

Subarray with Given Sum / XOR:

Subarray Sum = K	Prefix sum + hash	count += mp[prefix-k]	O(n)	O(n)
Count Subarrays XOR = K	Prefix XOR + hash	count += mp[prefix^k]	O(n)	O(n)
Longest Subarray Sum K	Prefix sum + hash	res = max(res,i-mp[prefix-k])	O(n)	O(n)
Longest 0/1 Equal Subarray	Convert 0→-1, prefix sum	Same as sum K	O(n)	O(n)

Sliding Window / Product / Min Size:

Maximum Product Subarray	DP	Track min/max (for negatives)	O(n)	O(1)
Minimum Size Subarray Sum	Sliding window	Shrink left while sum>=target	O(n)	O(1)
Count Subarrays Product < K	Sliding window	count += r-l+1 for window	O(n)	O(1)

Triplets & Container:

3Sum	Sort + Two Pointers	Skip duplicates	O(n ²)	O(1)
Container With Most Water	Two pointers	Move smaller height	O(n)	O(1)

Advanced / Misc:

Longest Mountain	Peak detection + expand left/right	Skip processed elements	O(n)	O(1)
------------------	------------------------------------	-------------------------	------	------