How to recognize which data structure to use in a Question (striver)

① ↳ Two Pointer / sliding window

↳ Array is sorted

→ Summation kinda problem (twosum, threesum)

→ by sorting we are not distorting the logic means array element position doesnt bother to get the optimal ans

② Sliding window

↳ Longest subarray → contiguoes ((−)is not there)
substring along with Hashing (map)

③ Hashing

↳ when you need to store the value like map the value or Like get Frequency.

↳ remember the past one in traversing time

→ using previous value in future basically.

④ Binary Search

↳ Find the minimum, or the maximum

→ It's not neccessary the array has to be sorted. (Book allocation, Aggressive cow)

↳ Monotonous increasing or decreasing ✓

→ Lower bound, upper bound (Range in given)

Stack and Queue trapping Rain water ★ ↑

★ ↳ monotonic stack (next greater element)

→ when you are standing at a point and you need to know which was the Last element that was (smaller / greater) than you, → NGE → Next greater ★
NSE → Next smaller ★

Queue

↳ Queue in graph.

LRU Cache ★ ★ ≪ LFU cache

↳ tree and graph (widely used)

Linked List

→

Bit Manipulation

↳ Power set ★

Recursion

↳ Out of all (min, max, print) All Possible ways. & try out all give me the best.

★★ → Constrains are very minimal (14, 15, 16 ...)
It has to be recursion.

All subset Problems / subsequence / Count No of ways

→ Pick and Not Pick

Backtracking

↳ Back tracking ≈ Recursion.

↳ N Queen

→ Rate in maze

→ Sudoku.solver

→ M colouring problem.

greedy Algo

→ Is there a pattern? there can never be a pattern.

→ Greedy means → when you read the problem whatever your heart says.

→ try in 10-15 ~~&&~~ test case.

Trees. (BST, BT)

↳ There are some standard Question (Striver (55) playlist)

Graphs

↳ Matrix → (rotten orange, Number islands)

Multisource BFS. ✓

→ shortest path algorithm. ✓

→ Disjoint set union. ✓ (Rank or path compressize (size))

→ MST (Minimum span tree)

→ Topologicalsort (Course ~~schdule~~ Schedule on LC)

Dynaamic Programming

↳ 1 D DP

→ 2 D DP

→ (actually check DP striver playlist)

all patterns are covered.

~~strings~~ trie

↳ XOR Problems (maximum)

→ Longest prefix or suffix (string one)

string

↳ pattern matching

→ Z / Rabbin cap / kmp

——— x ———