

How to recognize which data structure to use in a question (summary)

① Two Pointer / sliding window

- ↳ Array is sorted
- Summation kinda problem (two sum, three sum)
- by sorting we are not distorting the logic means array element position doesn't bother to get the optimal ans.

② Sliding window

- ↳ Longest subarray ^{contiguous} (— 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 necessary that array has to be sorted. (Book allocation, Aggressive cow)
- ↳ Monotonous increasing or decreasing ✓
- Lower bound, upper bound (Range is given)

Stack and Queue → trapping rain water (next greater element)

* 1) 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 than you), → NGE → Next greater +
NSE → Next smaller +

Queen

↳ Queue in graph.

LBU Cache \nrightarrow LFU cache

→ tree and graph (widely used)

Linked List

→

Bit Manipulation

↳ Power is $W \rightarrow$

Recursion

↳ Out of all (min, max, print) All Possible ways.

→ try out all give me the best.

→ try out all the
* → constraints are very minimal (14, 15, 16...)

It has to be recursion.

All subset problems / subsequence / count no of ways

→ Pick and Not Pick

Backtracking

↳ Backtracking \approx Recursion.

- ↳ N Queen
- Rat in maze
- Sudoku solver
- M colouring problem.

Greedy Algo

- In 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 → (color orange, Number islands)
Multisource BFS. ✓

- Shortest path algorithm. ✓
- Disjoint set union. ✓ (Rank or Path compressize) (size)
- MST (Minimum span tree)
- Topological sort (Course schedule on Le)

Dynamic Programming

↳ 1D DP

→ 2D DP

→ (actually check DP stinner playlist)
all patterns are covered.

~~strings~~ trie

↳ XOR Problems (maximum)

→ longest prefix or suffix (string one)

string

↳ pattern matching

→ Z / Rabin Karp / KMP

_____x_____