Gurtaj Singh
@gurtajs4

Intuition for **Solving** Coding Question in **Interview**

Follow for more updates

Intuition #1

If the given input is a **sorted array**, we will either be using **Binary Search** or **Two Pointers** strategy

Intuition #2

If the problem is related to a **LinkedList** and we can't use extra space, then use the **Fast** & **Slow Pointer** approach.

Intuition #3

If we are dealing with

top/maximum/minimum/closest

'K' elements among 'N' elements,

we will be using a **Heap**.

Intuition #4

If we need to try all **combinations** (or permutations) of the input, we can either use

Backtracking

or

Breadth First Search(BFS).

Intuition #5

Most of the questions related to **Trees** or **Graphs** can be solved either through **BFS** or **DFS**.

Intuition #6

Recursive solution can be converted to an **iterative** solution using a **Stack**.

Intuition #7

For a problem involving **arrays**,
if there exists a solution in
O(n^2) time and **O(1)** space,
there must exist two other solutions:

1) Using a **HashMap** or a **Set** for **O(n)** time and **O(n)** space,
2) Using **sorting** for **O(n log n)** time and O(1) space.

Intuition #8

If a problem is asking for **optimization** (e.g., **maximization** or **minimization**), we will be using **Dynamic Programming**.

Intuition #9

If we need to
find some **common substring**
among a set of strings,
we will be using a **HashMap** or
Trie.

Intuition #10

If we need to **search/manipulate** a bunch of strings, **Trie** will be the best data structure.