Today's Questions

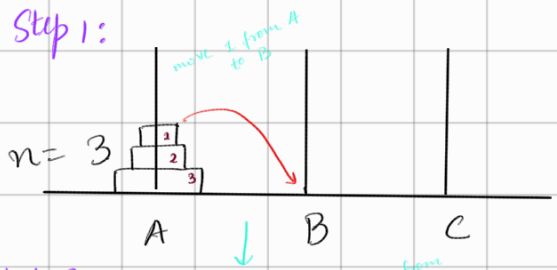
① Tower of Hanoi

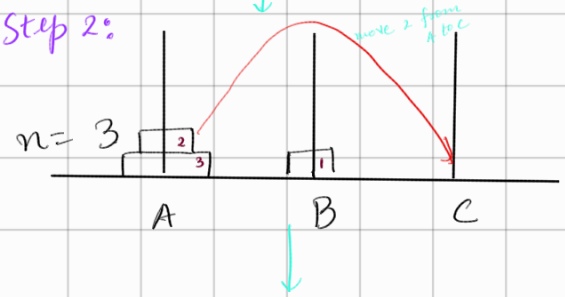
② All indices of array

+ Discussion on DSA based projects.

① Tower of Hanoi

Question: move all disks from tower A to tower B using tower C as helper

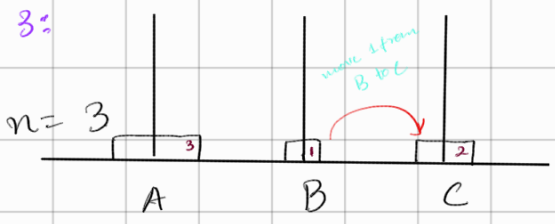


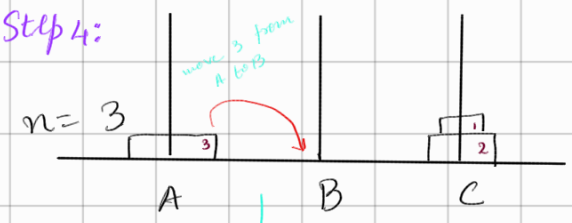


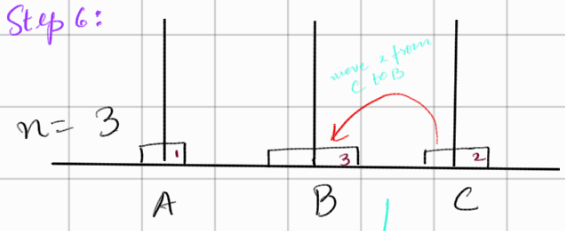
Constraints

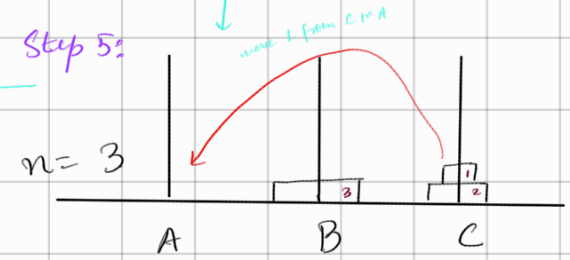
1) Only one disk can be moved at a time

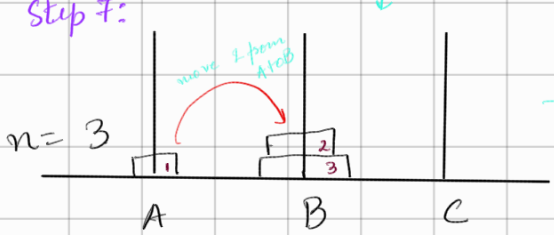
2) Larger disk cannot be placed on top of smaller disk

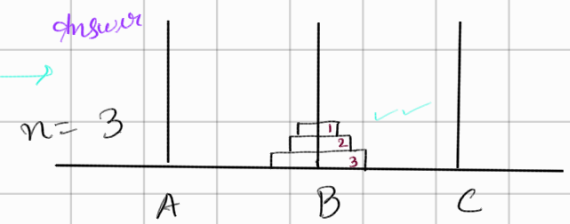












∴ Three main steps are

Steps 1, 2 & 3 ① move n-1 disks from A to C using B as helper

Step 4 ② move n from A to B

Steps 5, 6, & 7 ③ move n-1 disks from C to B using A as helper

∴ Here ① & ③ are faith for n-1 / smaller problem and ② is current work for nth problem.

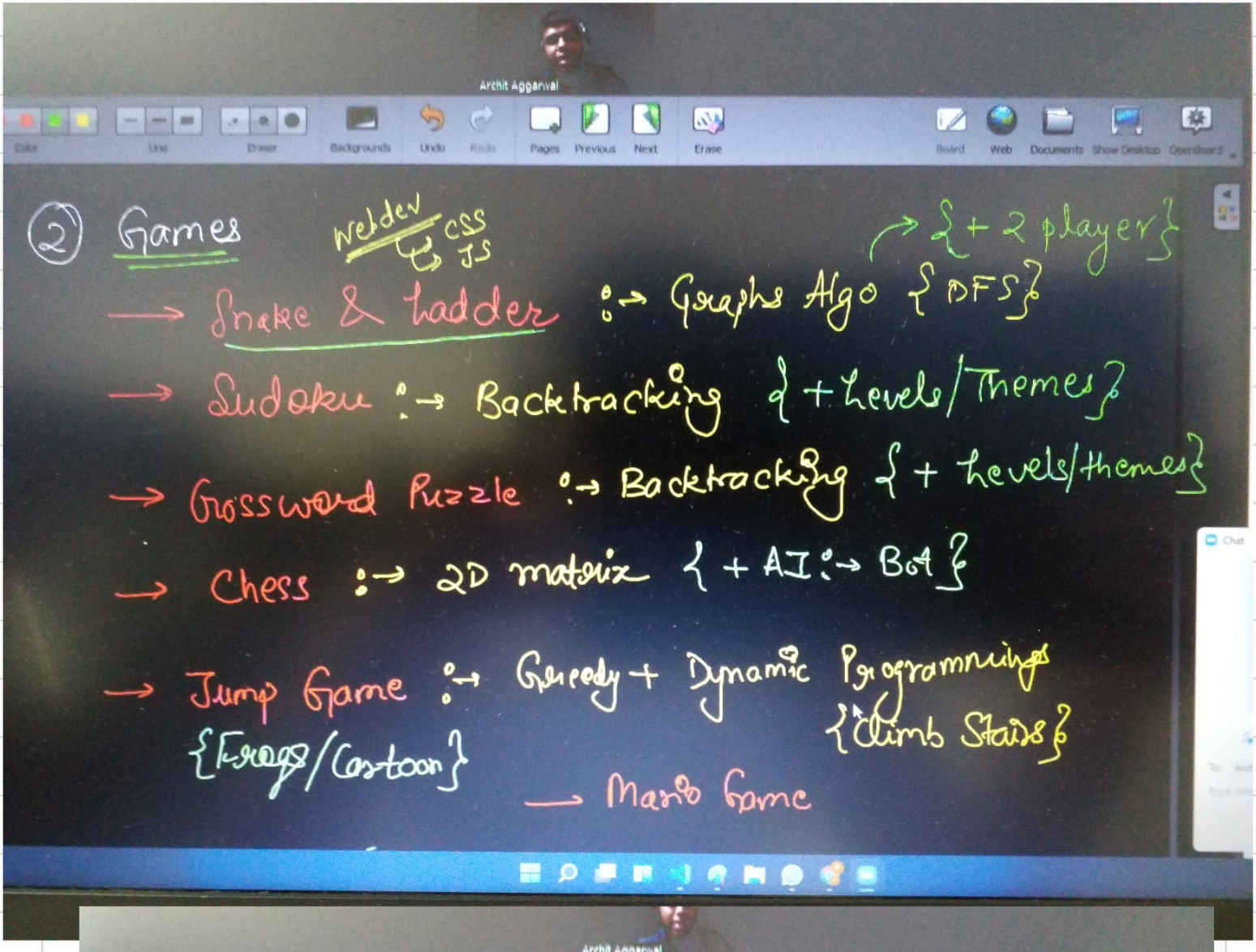
∴ CODE :

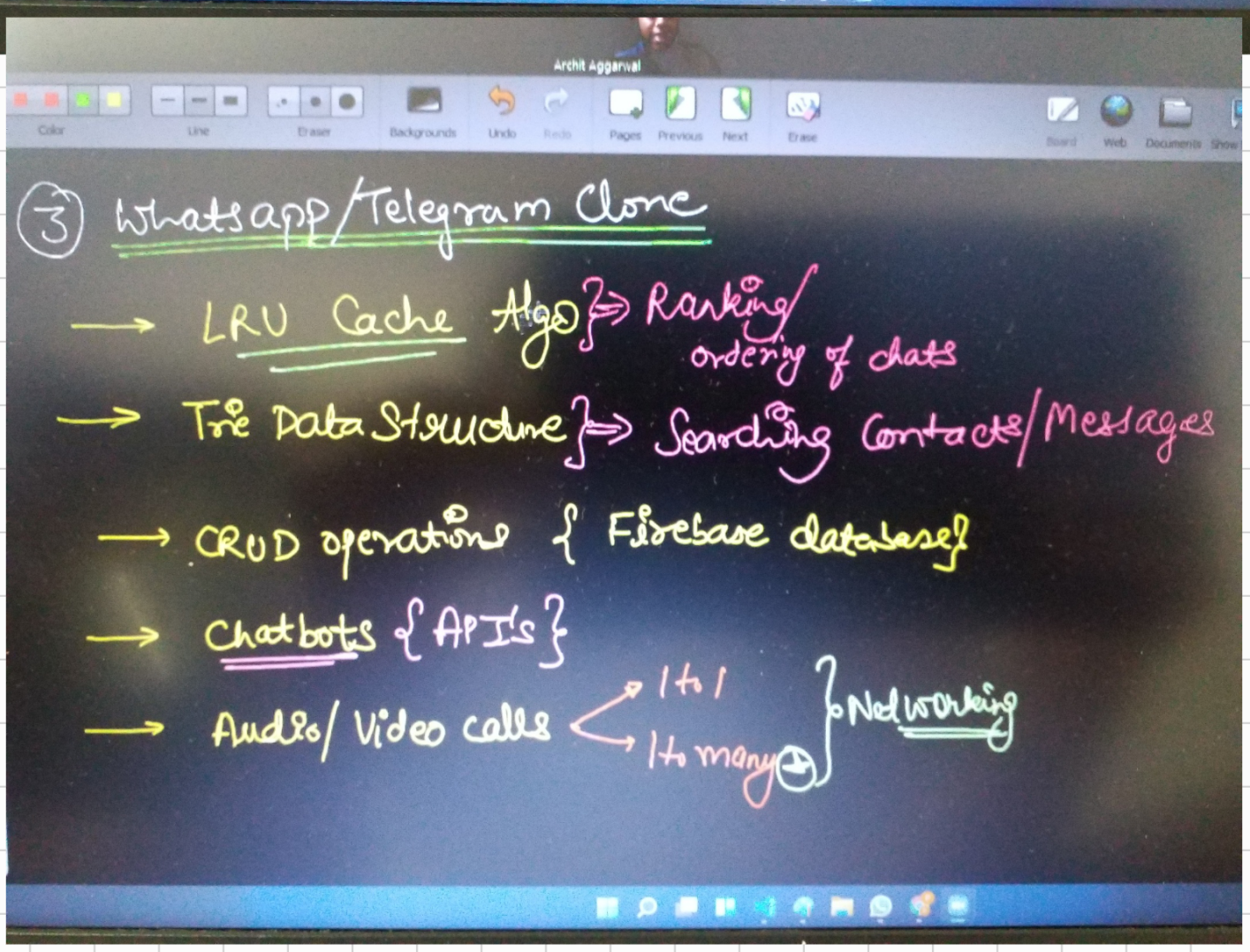
```
public static void toh(int n, int srcTower, int distTower,
                                        - int auxTower){
    if(n==0)  // no disks left
        return

    toh(n-1, srcTower, auxTower, distTower); // ①
    Syso(n + "[" + srcTower + " → " + distTower + "]"); // ②
    toh(n-1, auxTower, distTower, srcTower); // ③
}
```

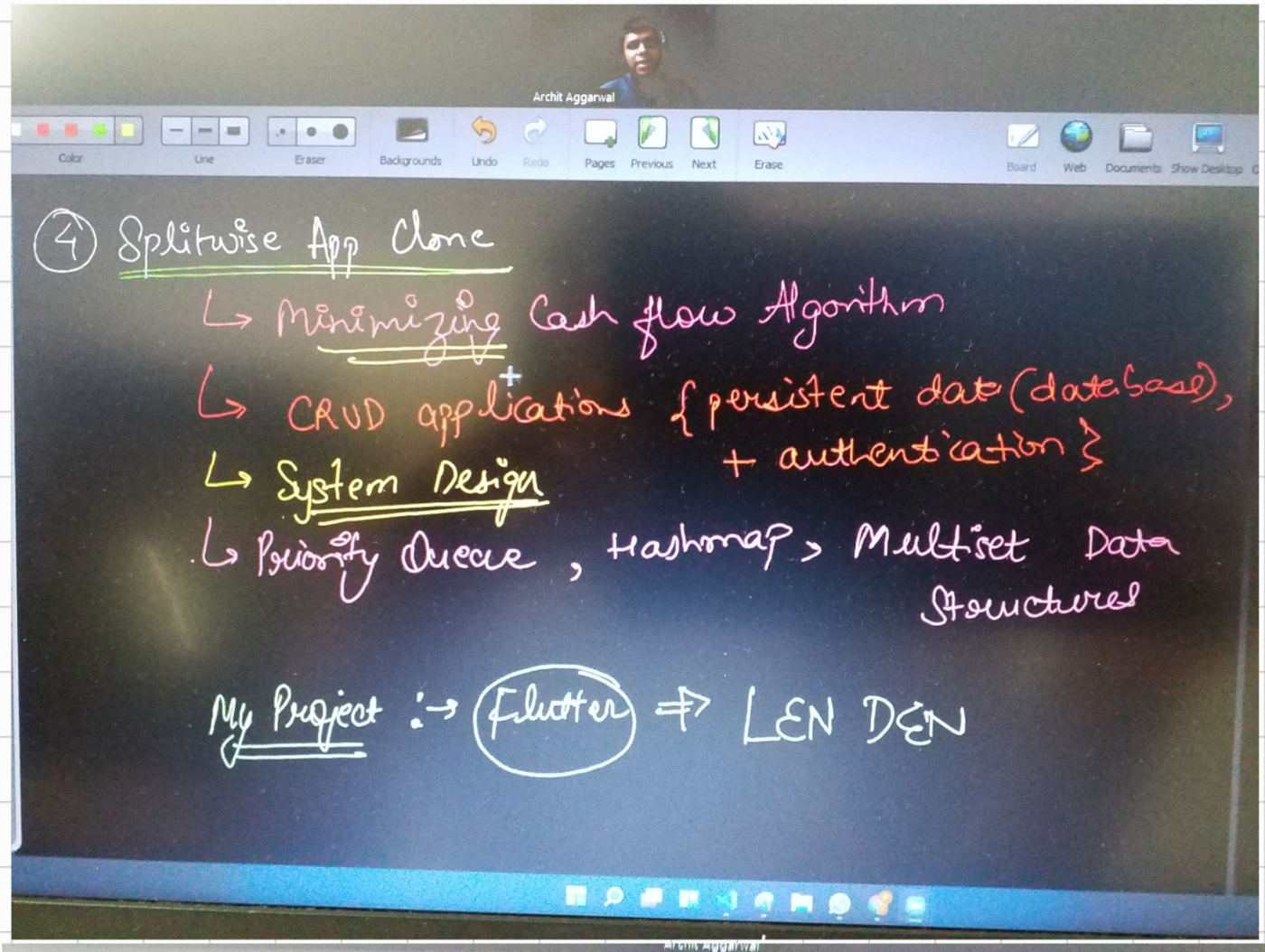
✳ DSA based projects {Unique Project Ideas}

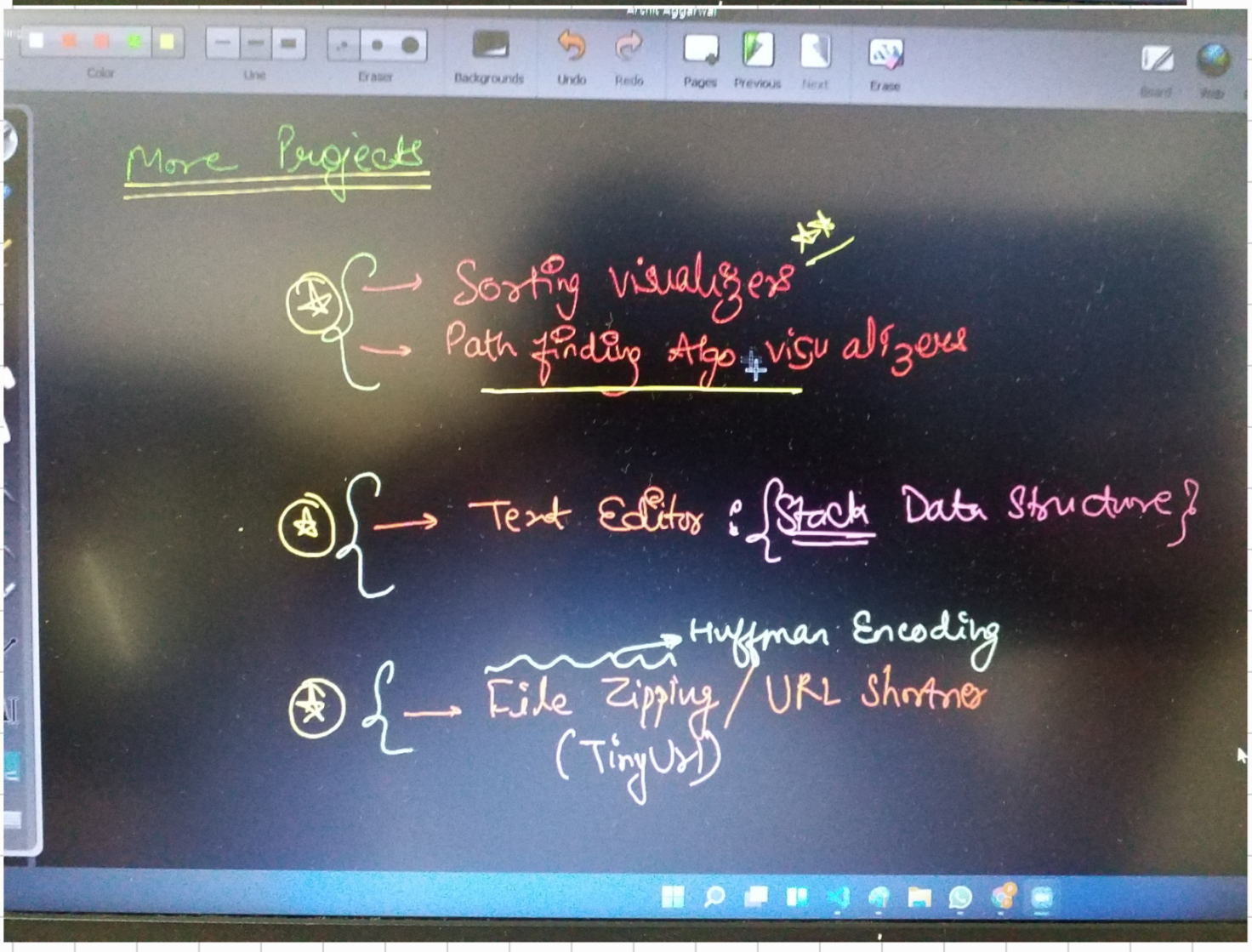
① writing own library / header file
→ B tree, B+ Tree, {SQL + DSA}
→ Segment Tree, Fenwick Tree, Number Theory Algorithms, DSU, Hashmap, Priority Queue/Heap, Graph Algorithms

{Competetive (Advanced DSA) + generic programming ⇓ Templates + OOPS}

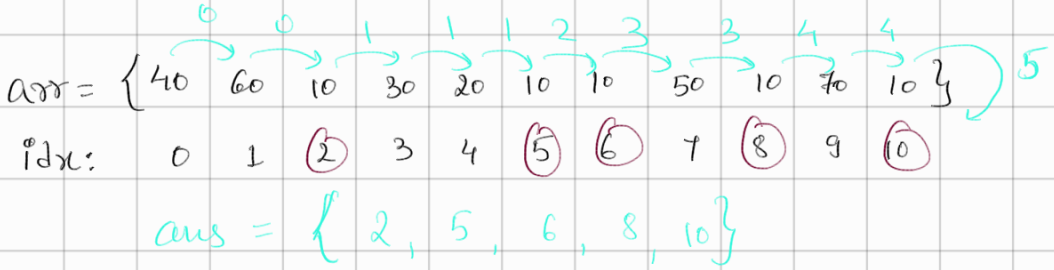

Archit Aggarwal
Color
Line
Eraser
Backgrounds
Undo
Redo
Pages
Previous
Next
Erase
Board
Web
Documents
Show Desktop
OpenBoard
(2) Games
Webdev
CSS
JS
→ Snake & ladder :→ Graphs Algo {DFS} {+2 player}
→ Sudoku :→ Backtracking {+levels/Themes}
→ Crossword Puzzle :→ Backtracking {+ levels/themes}
→ Chess :→ 2D matrix {+AI :→ Bot}
→ Jump Game :→ Greedy + Dynamic Programming {Climb Stairs}
{Frogs/Cartoon}
→ Mario Game
Chat


Archit Aggarwal
Color
Line
Eraser
Backgrounds
Undo
Redo
Pages
Previous
Next
Erase
Board
Web
Documents
Show
(3) Whatsapp/Telegram Clone
→ LRU Cache Algo} ⇒ Ranking/ ordering of chats
→ Trie Data Structure} ⇒ Searching Contacts/Messages
→ CRUD operations {Firebase database}
→ Chatbots {APIs}
→ Audio/Video calls → 1 to 1
→ 1 to many} Networking


Archit Aggarwal
Color
Line
Eraser
Backgrounds
Undo
Redo
Pages
Previous
Next
Erase
Board
Web
Documents
Show Desktop
④ Splitwise App Clone
↳ Minimizing Cash flow Algorithm
↳ CRUD applications {persistent data (database), + authentication}
↳ System Design
↳ Priority Queue, Hashmap > Multiset Data Structures
My Project :→ Flutter ⇒ LEN DEN


Color
Line
Eraser
Backgrounds
Undo
Redo
Pages
Previous
Next
Erase
More Projects
Sorting Visualizers
Path finding Algo visualizers
Text Editor : {Stack Data Structure}
Huffman Encoding
File Zipping / URL Shortner
(TinyUrl)

② All Indices of Array



Preorder work

⟹ We cannot know size of answer array, until we traverse the whole array

→ when we are at base case (index == arr.length)
now we know size of answer
return new int[countof1];

// when returning Postorder Work

⟹ for any index idx, if element at idx == x then

ans[countof10] = idx;

else

return ans; // without updating

CODE:

```
if(idx >= arr.length)
    return new int[count];
if(arr[idx] == target) {
    int[] res = allIndices(arr, target, idx+1, count+1);
    res[count] = idx;
} else {
    int[] res = allIndices(arr, target, idx+1, count);
    return res;
}
```