

Mahender Kumar (mkumar04)

03:37

Take control

Pop out

Dial pad

Hold

Transfer

Chat

People

View

Apps

More

Camera

Mic

Share

Leave

Pwd.tx • - Exch • Harish • build_packag httpM • CWindows5y -ADAL • reports.cpp autotest.cpp Release2Proj Best Pi • InTrust_11

File Edit View

CR-84266 -

OpenSSL Changes+Pwd changes+Version changes+FIPS migratool changes b

Ln 101, Col 1 1,150 characters 90% Windows (CRLF) UTF-8

Mahender Kumar (mkumar04)

HK

Windows taskbar with icons for Edge, File Explorer, Chrome, Teams, and other applications.

System tray showing time 15:32, date 27-01-2025, and language ENG.

Mahender Kumar (mkumar04)

03:41

Take controlPop outDial padHoldTransferChatPeopleViewAppsMoreCameraMicShareLeave

prodcloudoutlook-my.sharepoint.com/:x/r/personal/mahender_kumar_quest_com/_layouts/15/doc.aspx?sourcedoc=%7B9A8C11E...

BXAQuestSprints - Status Rep...InTrust - Code Quali...WSRRequest Access - Co...InTrust builder in th...Simplifying Window...Getting error while...

11.6.2 QA Builds

Search for tools, help, and more (Alt + Q)

FileHomeInsertSharePage LayoutFormulasDataReviewViewAutomateHelpDraw

B7

ID's	Dev Owner	Developemt	Status	Build No	Build Date	QA Owner	QA Status	Commer
Version Update InTrust 11.6.2	Harish	Done		CopyRightVersion_11.6.2/11.6.2.9906			Pending	Dev: We just changed the the intrust pr Dev: We add some co password . For reference details
d Strength Validation in InTrust Suite Setup	Krishan	Done		PasswordValidation/11.6.2.9934			Pending	Dev: We made some char where we add the noti KBArticle
t for Upgrade Including tools	Krishan /Harish	Done		Fips-Migrator-Tool-11.6.2/11.6.2.9936 FE_OpenSSLUpdate3_4_0/InTrustPackage_11.6.1			Pending	Dev: We upgrade the Op both platfo
penSSL vulnerability (CVE 2024 5535)	Mahender/Krishan	Done		.9942.exe			In-Progress	DEV: We upgrade the Op both platforms + Docum Agent_memory
penSSL vulnerability (CVE 2024 5535)	Krishan	Done		FE_OpenSSLUpdate3_4_0/ InTrustPackage_11.6.1.9974.exe			In-Progress	

11.6.2 QA Builds

Workbook Statistics

Mahender Kumar (mkumar04)

3:32 PM

1/27/2025

Mahender Kumar (mkumar04)

HK

15:32

27-01-2025

L12: Print all Permutations of a String/Array | Recursion | Approach - 1

5. Subset-II

https://www.youtube.com/watch?v=Rln3gOkbhQE&list=PLgUwDviBlf0p4ozDR_kJJkONnb1wdx2Ma&index=53

6. K-th permutation Sequence

https://www.youtube.com/watch?v=wT7gcXLYoao&list=PLgUwDviBlf0p4ozDR_kJJkONnb1wdx2Ma&index=55

Day10: (Recursion and Backtracking)

1. Print all Permutations of a string/array
2. N queens Problem
3. Sudoku
4. M coloring Problem (Graph prob)
5. Rat in a Maze
6. Word Break (print all ways)

Day11: (Divide and Conquer)

1. 1/N-th root of an integer (use binary search) (square root, cube root, ..)
2. Matrix Median
3. Find the element that appears once in sorted array, and rest element appears twice (Binary search)
4. Search element in a sorted and rotated array/ find pivot where it is rotated
5. Median of 2 sorted arrays
6. K-th element of two sorted arrays

L12: Print all Permutations of a String/Array | Recursion | Approach - 1

5. Subset-II

https://www.youtube.com/watch?v=RIn3gOkbhQE&list=PLgUwDviBIf0p4ozDR_kJJkONnb1wdx2Ma&index=53

6. K-th permutation Sequence

https://www.youtube.com/watch?v=wT7gcXLYoao&list=PLgUwDviBIf0p4ozDR_kJJkONnb1wdx2Ma&index=55

Day10: (Recursion and Backtracking)

1. **Print all Permutations of a string/array**
2. N queens Problem
3. Sudoku
4. M coloring Problem (Graph prob)
5. Rat in a Maze
6. Word Break (print all ways)

Day11: (Divide and Conquer)

1. 1/N-th root of an integer (use binary search) (square root, cube root, ..)
2. Matrix Median
3. Find the element that appears once in sorted array, and rest element appears twice (Binary search)
4. Search element in a sorted and rotated array/ find pivot where it is rotated
5. Median of 2 sorted arrays
6. K-th element of two sorted arrays

Settings

TUF

0:13 / 19:06

46. Permutations of a String/Array | Recursion | Approach - 1

Description

Solution

Discuss (999+)

Submissions

46. Permutations

Medium

5534

128

Add to List

Share

Given an array `nums` of distinct integers, return *all the possible permutations*. You can return the answer in any order.

Example 1:

Input: `nums = [1,2,3]`

Output: `[[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]`

Example 2:

Input: `nums = [0,1]`

Output: `[[0,1],[1,0]]`

Example 3:

Input: `nums = [1]`

Output: `[[1]]`

Settings

TUF



0:29 / 19:06



[1, 2, 3]

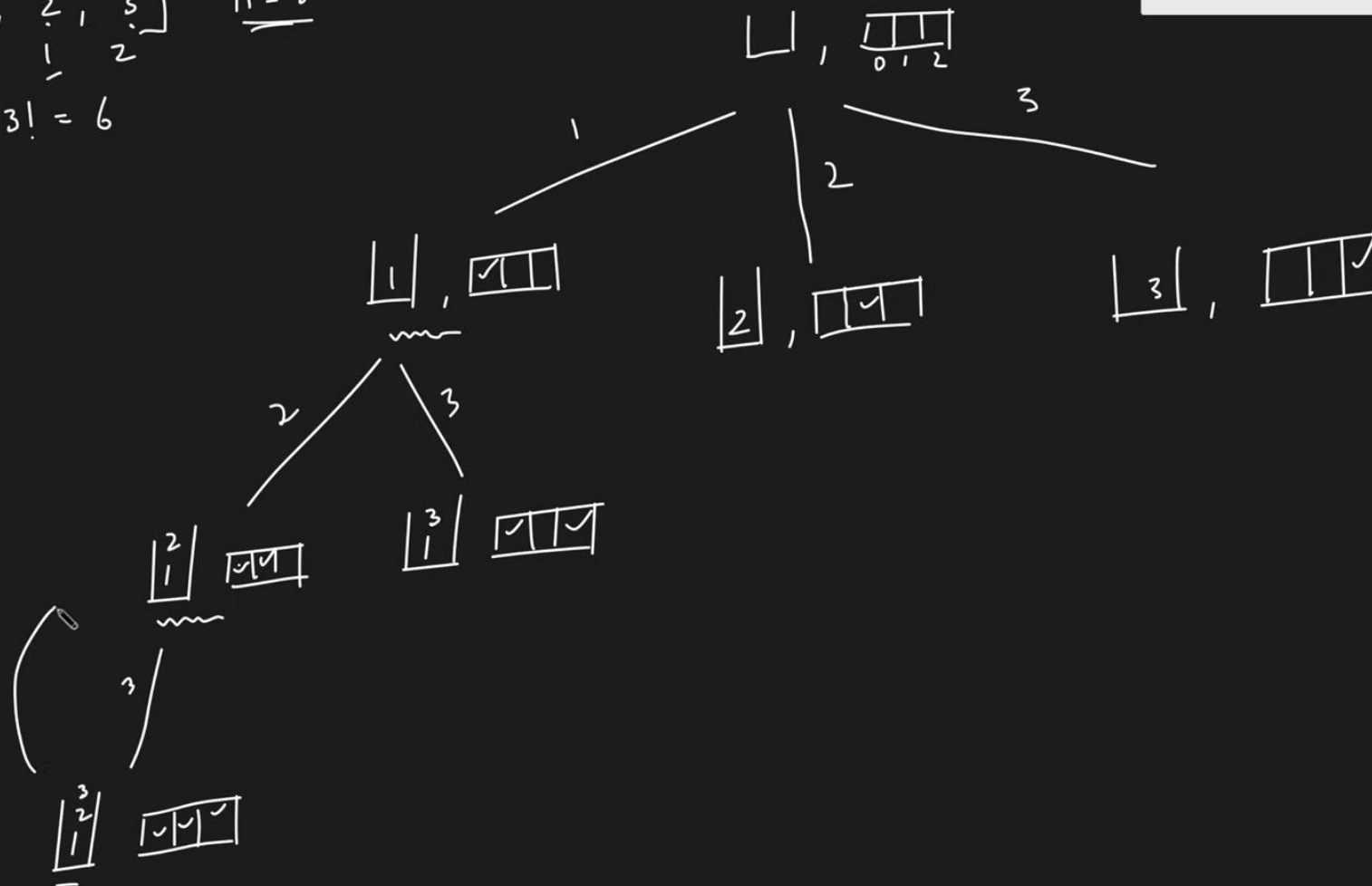
Let's give back to the community :)



$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \end{bmatrix} \quad n=3$$

$$n! = 3! = 6$$

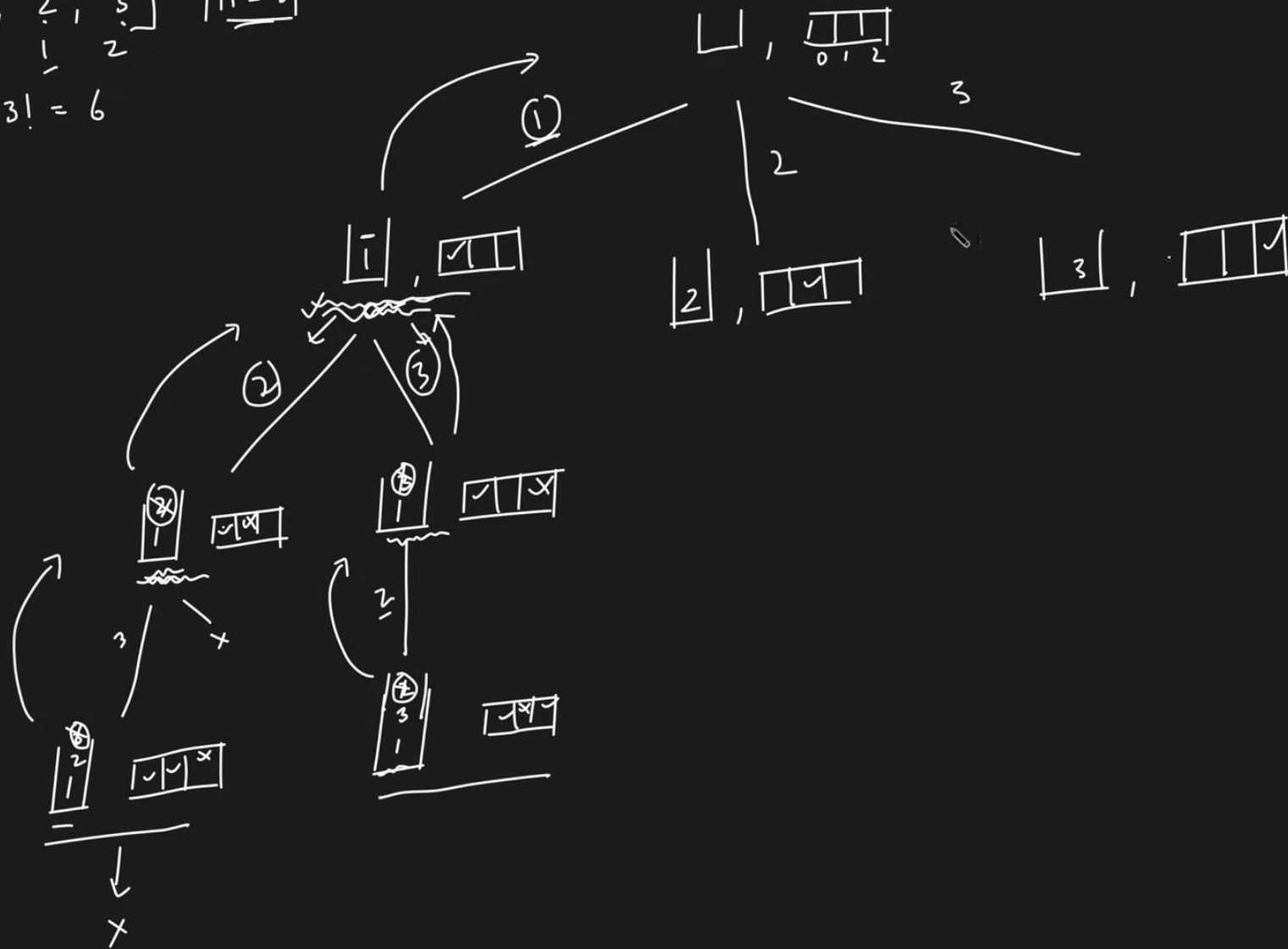
1, 2, 3



$$\left[\begin{array}{ccc} \checkmark & \downarrow & \\ \frac{1}{0} & \frac{2}{1} & \frac{3}{2} \end{array} \right] \quad |n=3|$$

$$n! = 3! = 6$$

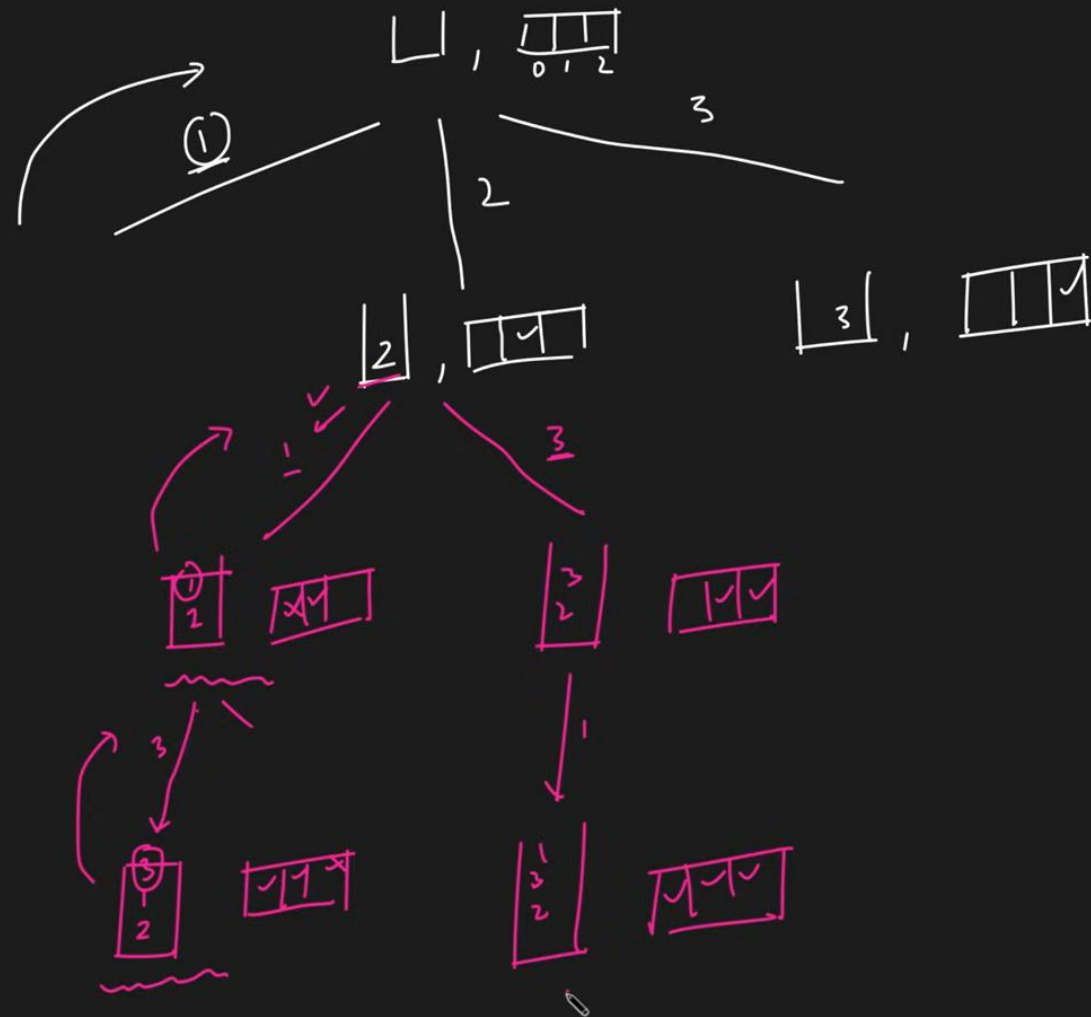
1, 2, 3
1, 3, 2



$$\left[\begin{array}{ccc} \checkmark & \downarrow & \\ \frac{1}{0} & \frac{2}{1} & \frac{3}{2} \end{array} \right] \quad \boxed{n=3}$$

$$n! = 3! = 6$$

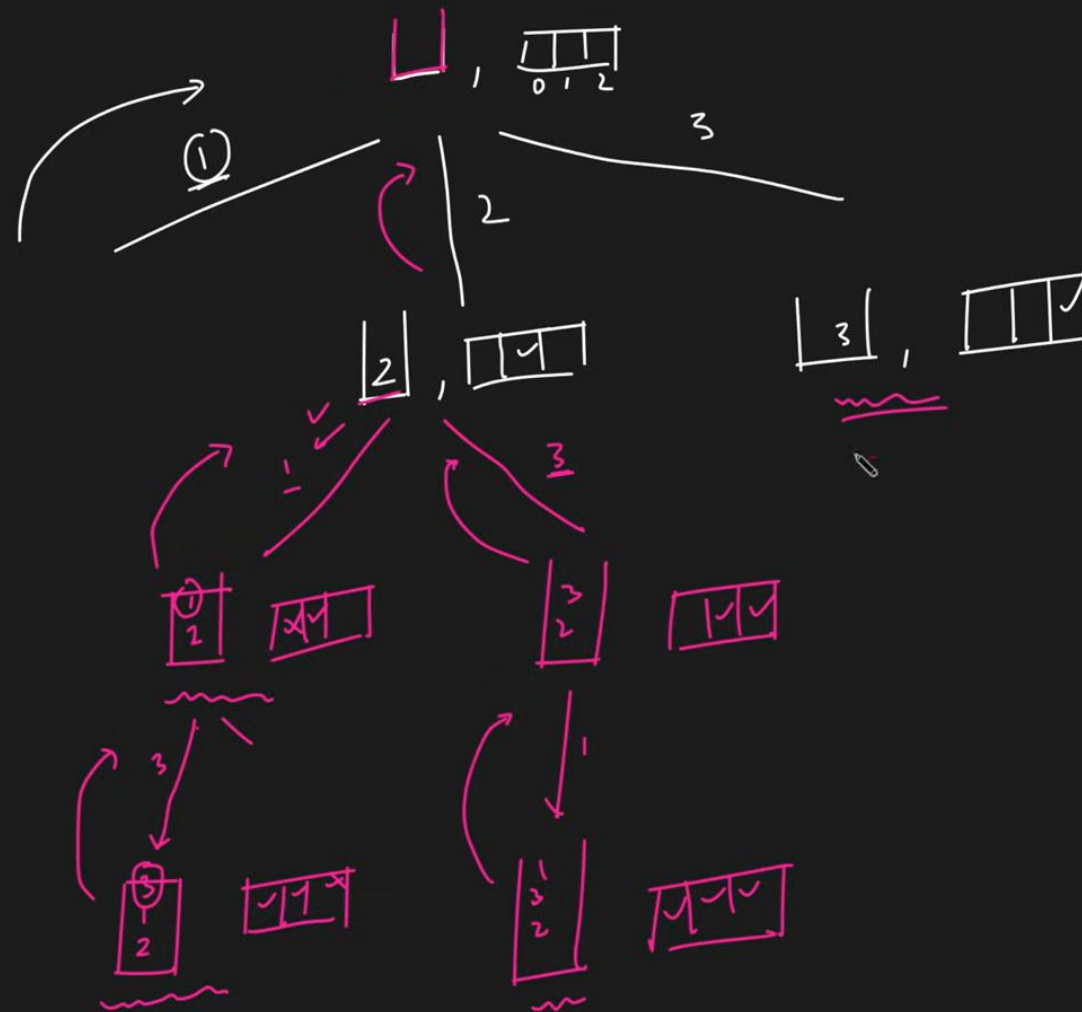
$1, 2, 3$
 $1, 3, 2$
 $2, 1, 3$
 $2, 3, 1$



$$\left[\begin{array}{ccc} \checkmark & \downarrow & \\ \frac{1}{0} & \frac{2}{1} & \frac{3}{2} \end{array} \right] \quad |n=3|$$

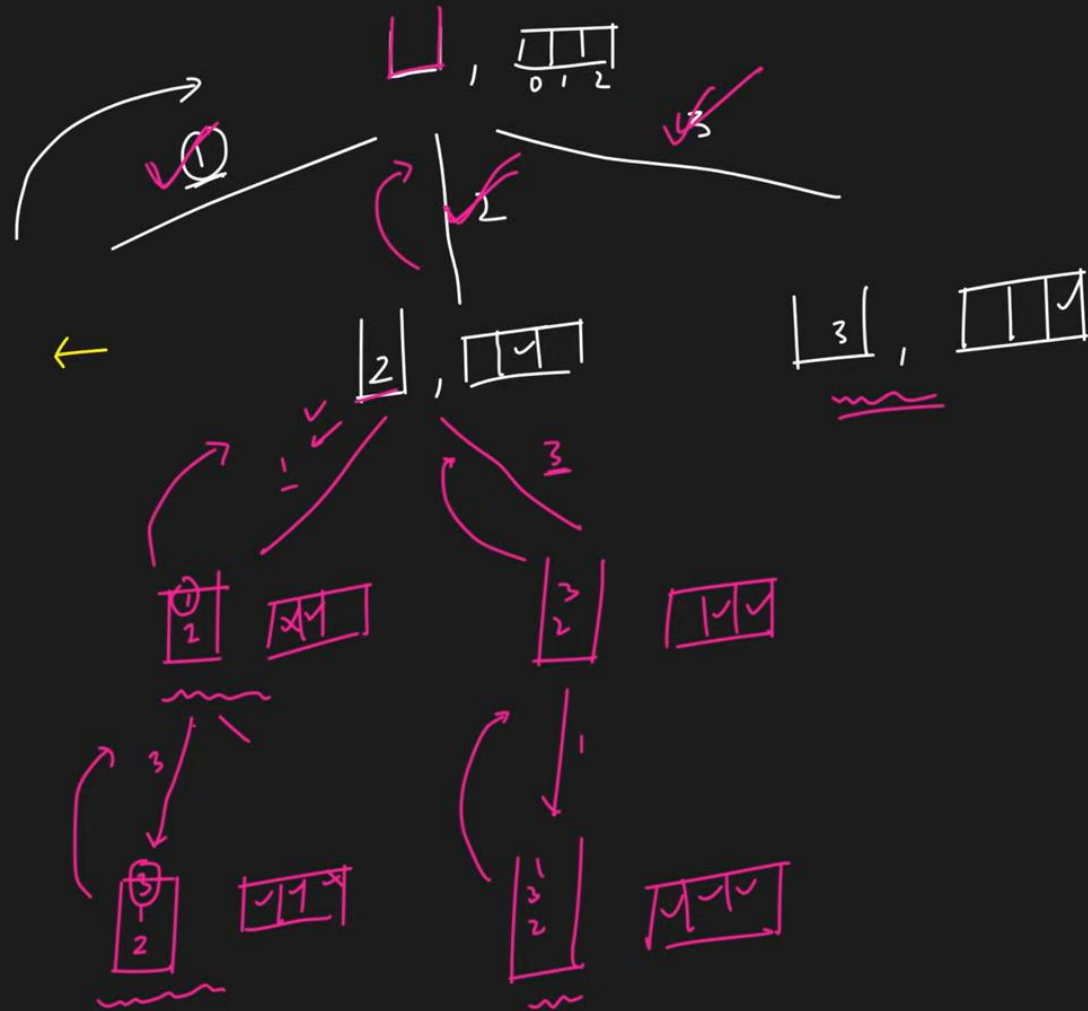
$$n! = 3! = 6$$

$1, 2, 3$
 $1, 3, 2$
 $2, 1, 3$
 $2, 3, 1$



$$\begin{bmatrix} \checkmark & \downarrow & \\ 1 & 2 & 3 \\ 0 & 1 & 2 \end{bmatrix} \quad | \underline{n=3}$$

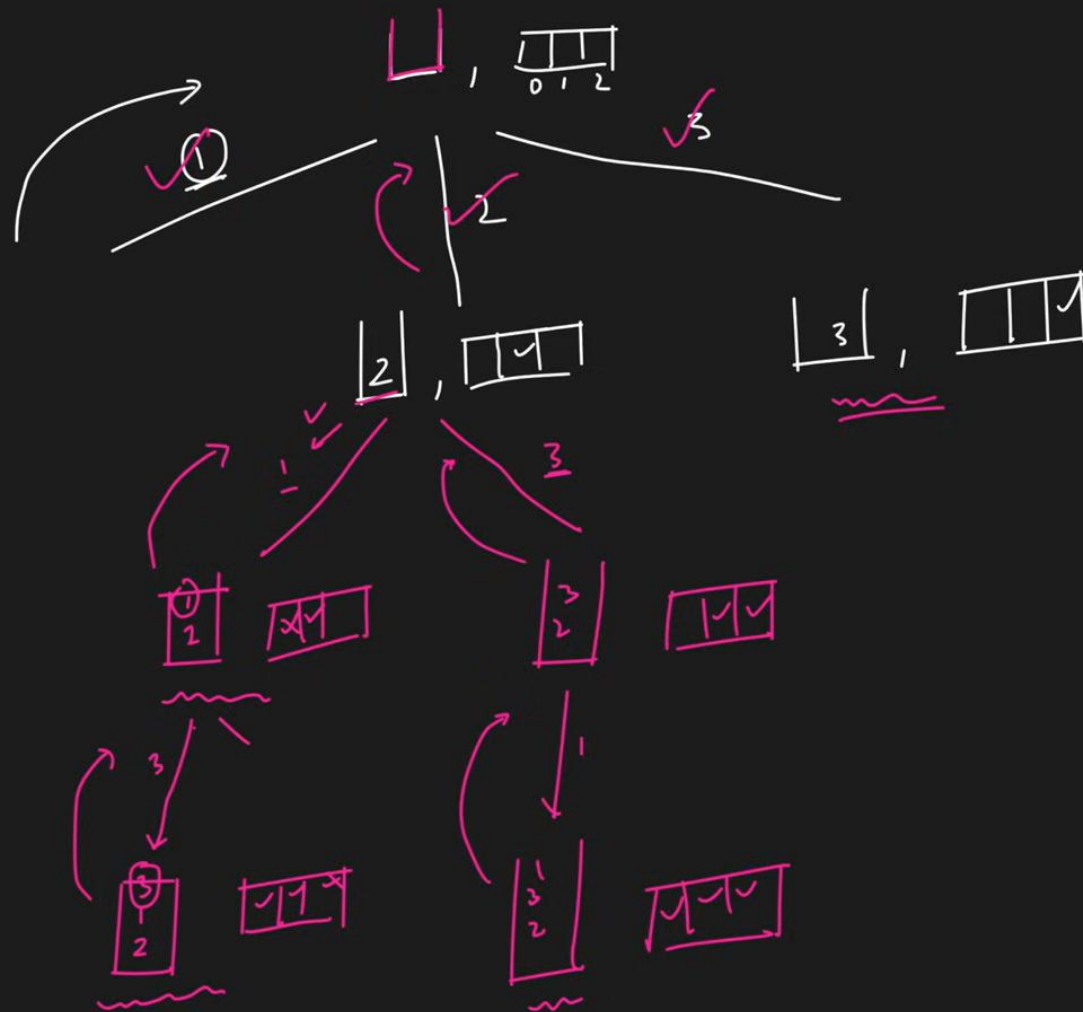
$$n! = 3! = 6$$



$$\left[\begin{array}{ccc} \checkmark & \downarrow & \\ \frac{1}{0} & \frac{2}{1} & \frac{3}{2} \end{array} \right] \quad \boxed{n=3}$$

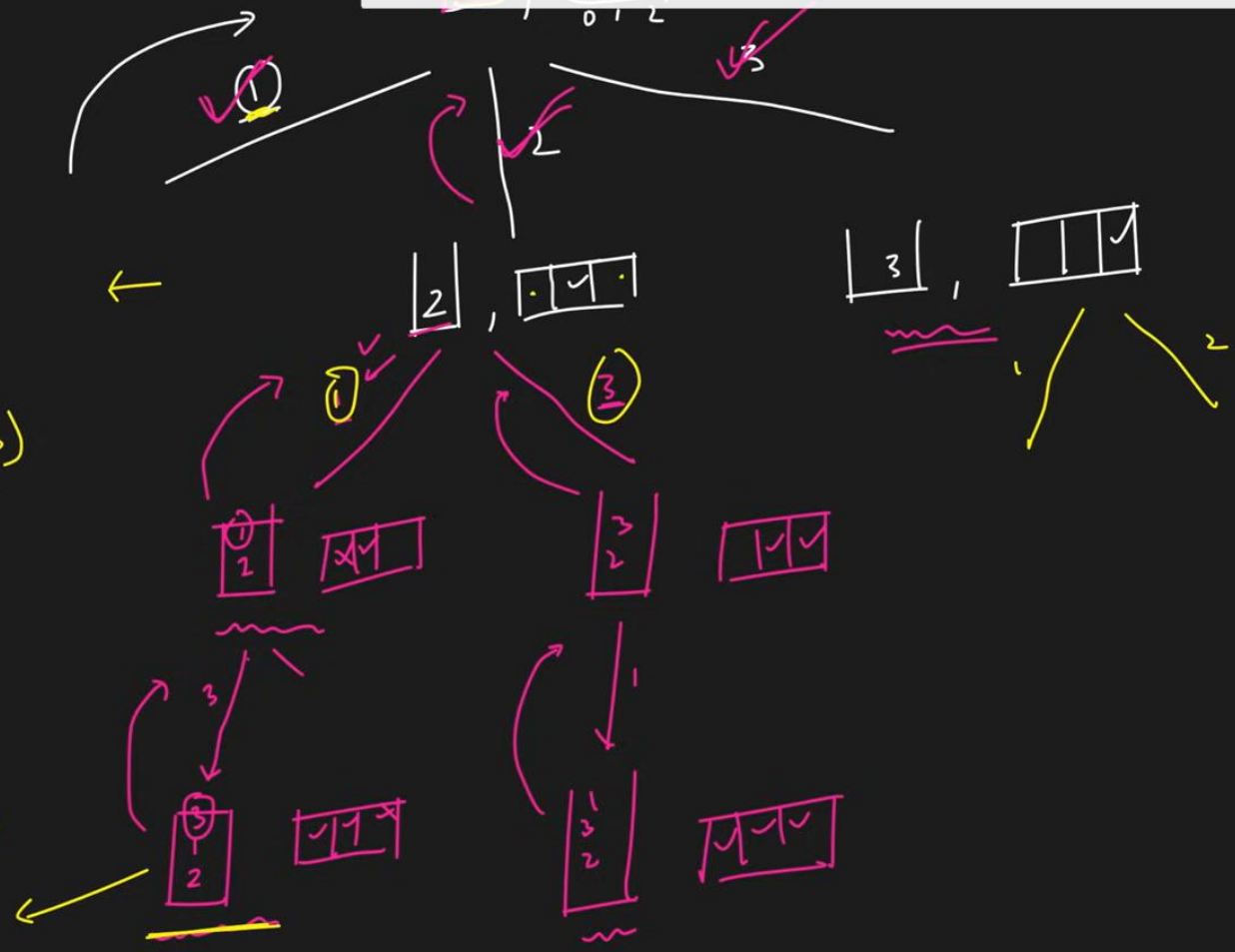
$$n! = 3! = 6$$

$$\left[\begin{array}{l} 1, 2, 3 \\ 1, 3, 2 \\ 2, 1, 3 \\ 2, 3, 1 \\ 3, 1, 2 \\ 3, 2, 1 \end{array} \right] \quad \left. \begin{array}{l} \\ \\ \\ \\ \\ \end{array} \right\} \begin{array}{l} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 0 \end{array}$$



$\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ 2 \end{bmatrix} \mid n=3$
 $n! = 3! = 6$

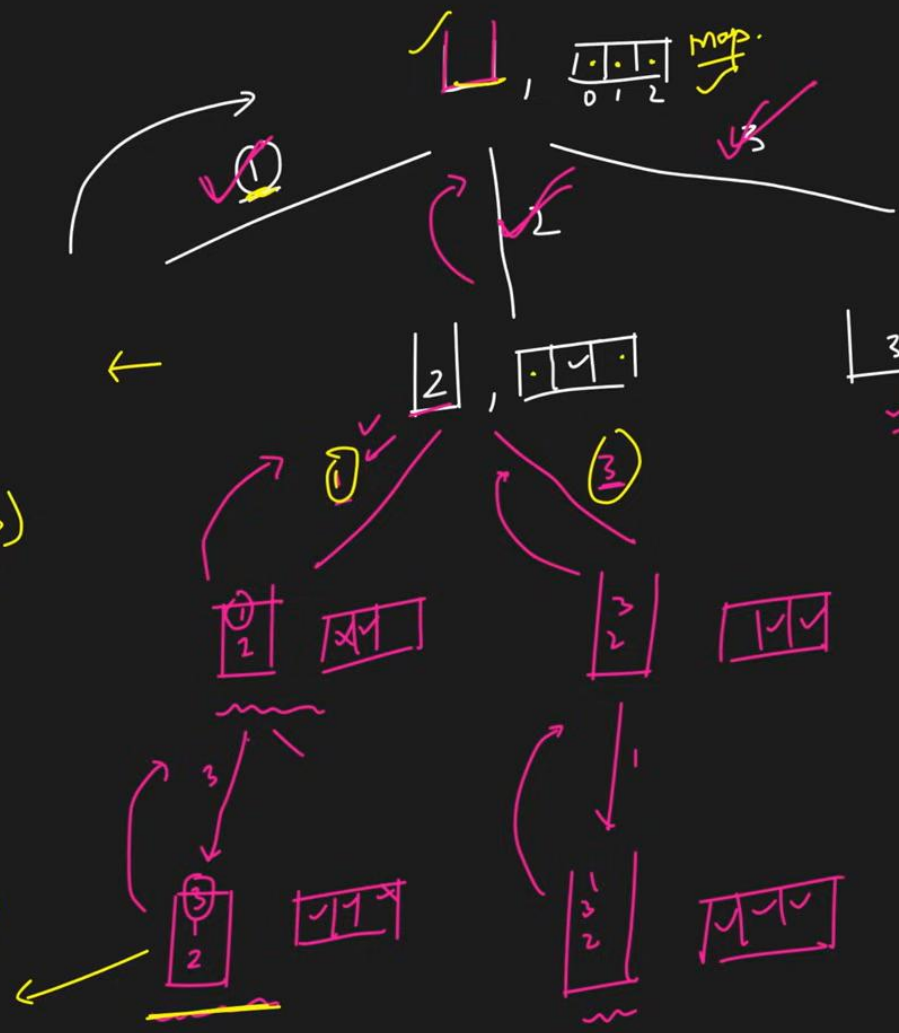
$f(ds, map)$
 $\left\{ \begin{array}{l} ds.add(a[i]) \\ map[i] = 1 \end{array} \right\} \text{ loop } (0 - n-1)$
 $if (i \text{ is } !map)$
 $f(ds, map)$
Ans.
 $(ds.size == n)$



$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ 2 \end{bmatrix} \quad | \quad n=3$$

$$n! = 3! = 6$$

$f(ds, map)$
 $\left\{ \begin{array}{l} ds.add(a[i]) \\ map[i] = 1 \end{array} \right\}$ loop $(0 - n-1)$
 $if (i \text{ is } !map)$
 $f(ds, map)$
 $ans.$
 $(ds.size == n)$



$T.C. \rightarrow n! \times n.$
 $SC \rightarrow O(n) + O(n)$

$$\left[\begin{array}{ccc} \checkmark & & \downarrow \\ \frac{1}{0} & \frac{2}{1} & \frac{3}{2} \end{array} \right] \quad | \underline{n=3}$$

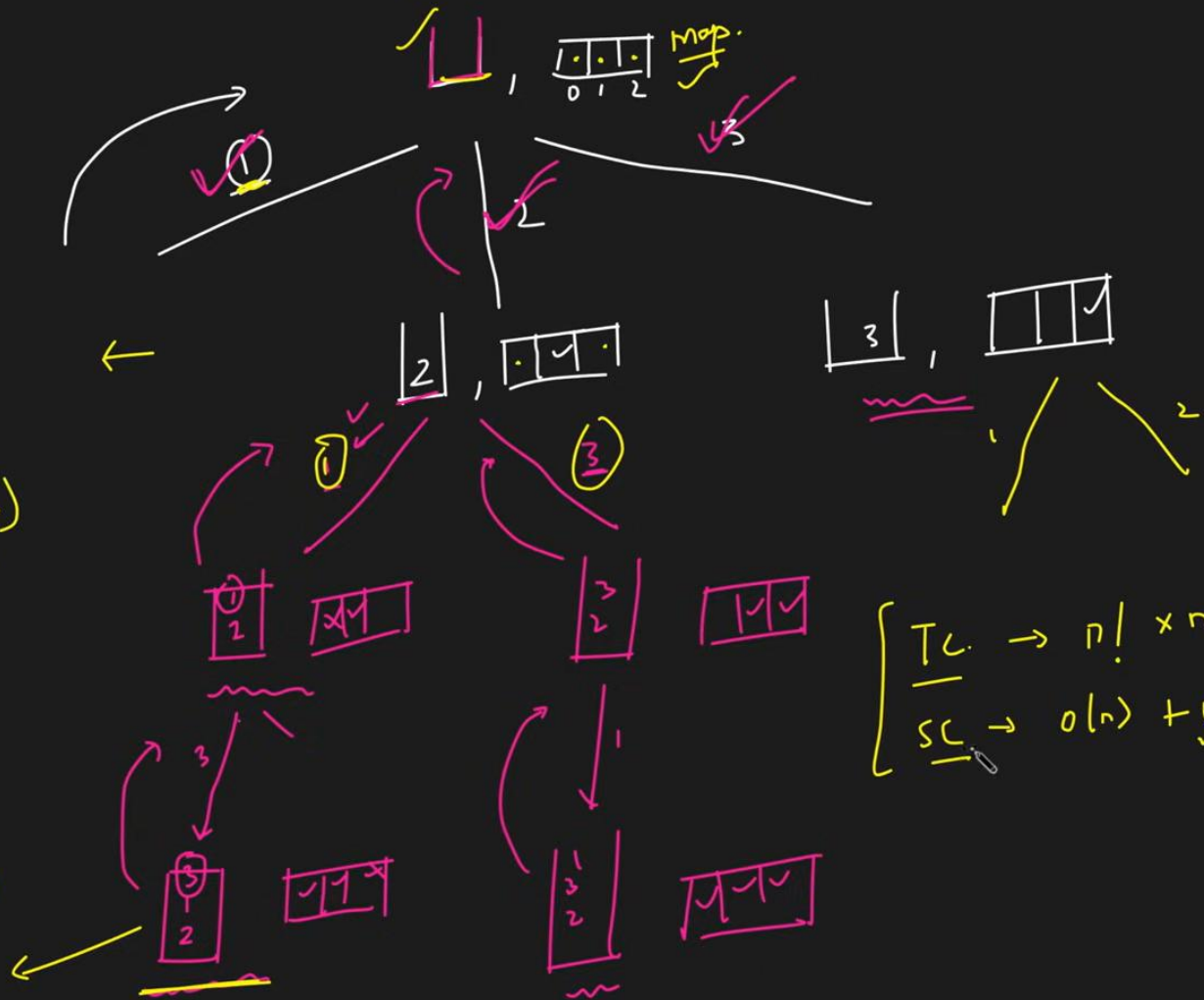
 $f(ds, map)$

$\left\{ \begin{array}{l} \text{do. add}(a[i]) \\ \text{map}[i] = 1 \end{array} \right\}$

$$g(ds, \text{mg})$$

Ans.

(des. size = n)



$$\left[\begin{array}{l} \underline{TC} \rightarrow n! \times n. \\ \underline{SC} \rightarrow o(n) + \underline{o(n)} \end{array} \right]$$

$$\left[\begin{array}{ccc} \checkmark & \downarrow & \\ \frac{1}{0} & \frac{2}{1} & \frac{3}{2} \end{array} \right] \quad | \quad \underline{n=3}$$

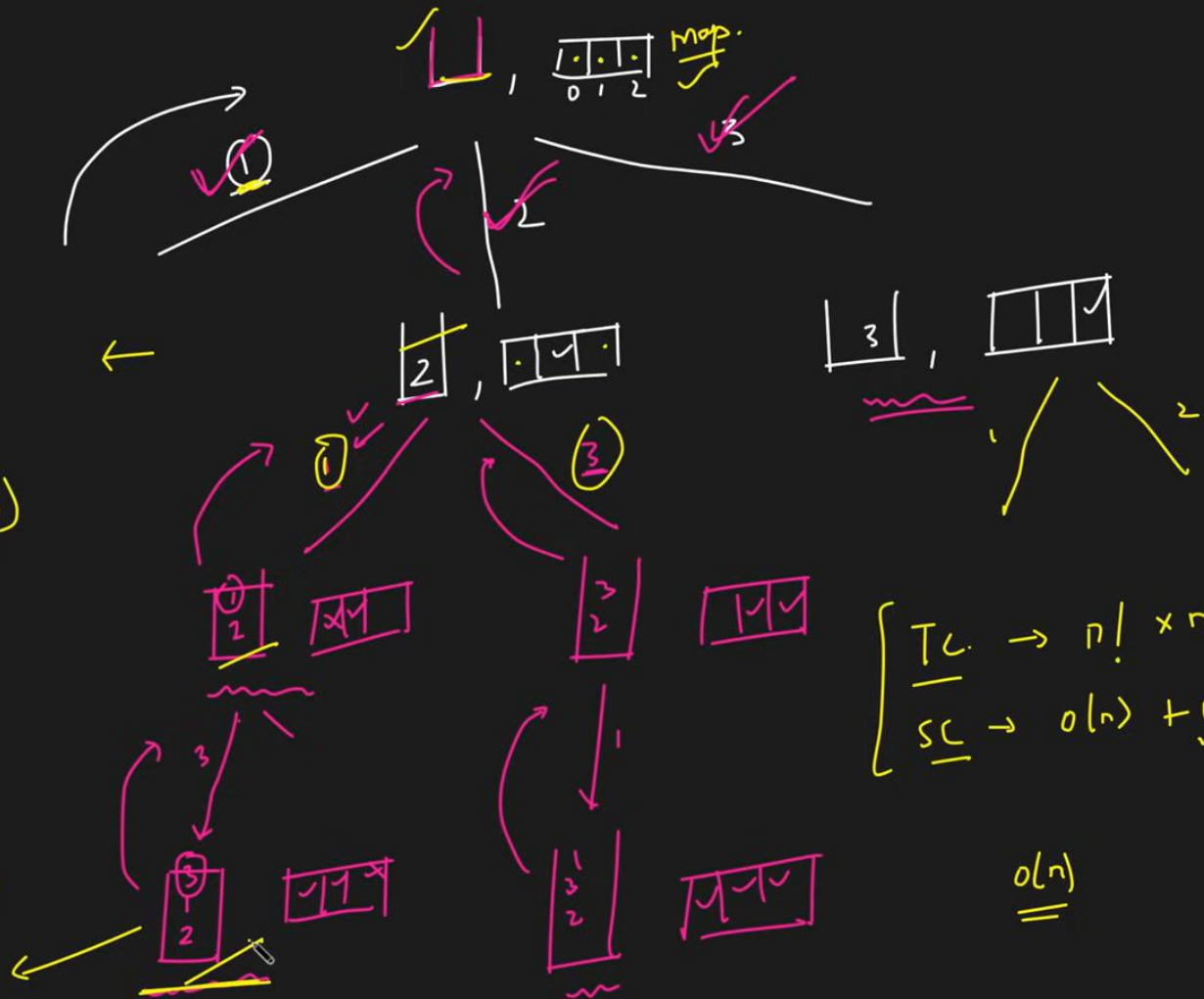
$$f(ds, map)$$

$\left\{ \begin{array}{l} \text{do. add}(a[i]) \\ \text{map}[i] = 1 \end{array} \right\}$

$$f(ds, \nabla f)$$

Ques.

(data size = n)



$$\left[\begin{array}{l} \underline{TC} \rightarrow n! \times n. \\ \underline{SC} \rightarrow O(n) + \underline{O(n)} \end{array} \right]$$

$$\underline{\underline{O(n)}}$$

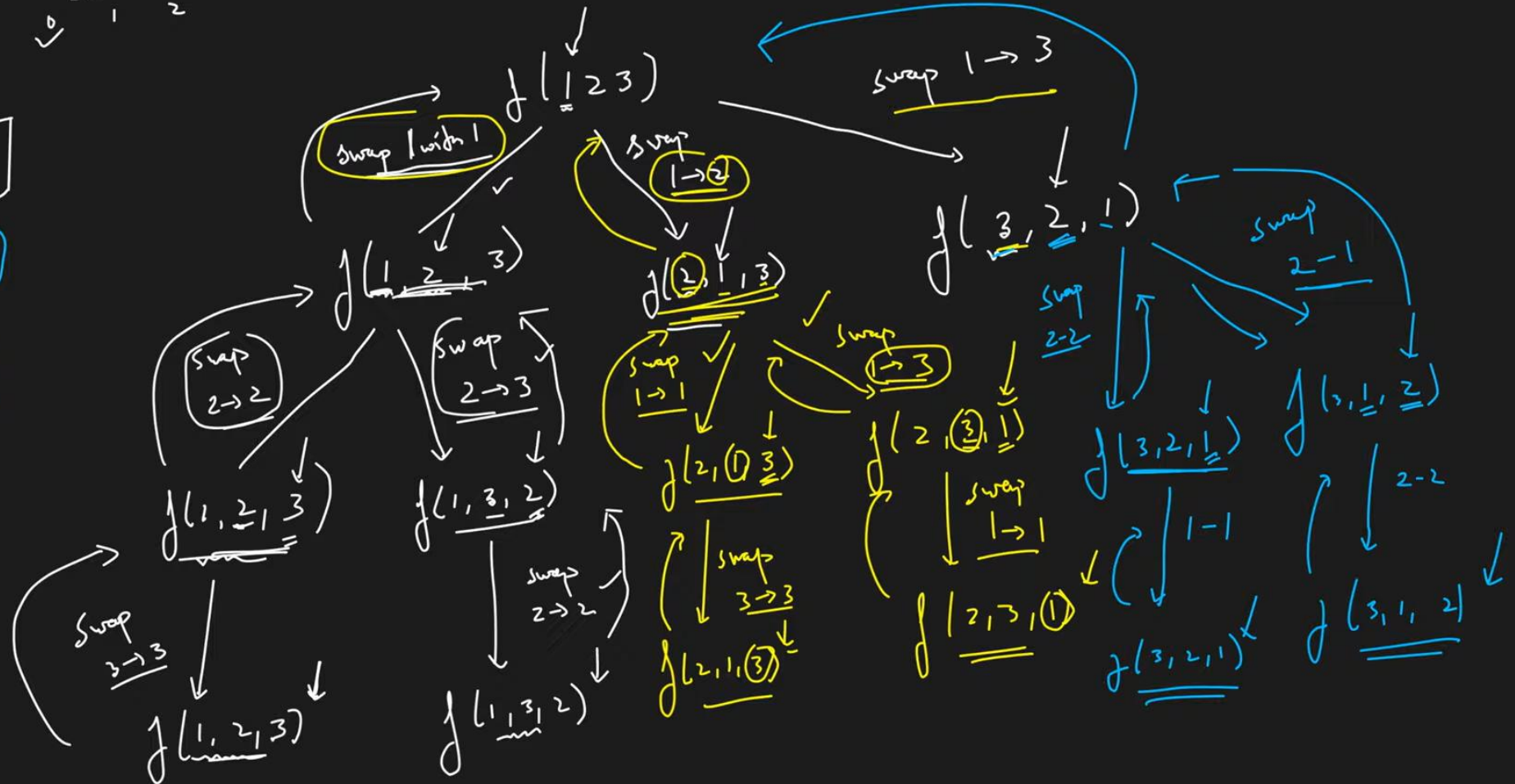
```
1 class Solution {  
2     private void recurPermute(int[] nums, List<Integer> ds, List<List<Integer>> ans, boolean []freq) {  
3         if(ds.size() == nums.length) {  
4             ans.add(new ArrayList<>(ds));  
5             return;  
6         }  
7         for(int i = 0; i < nums.length; i++) {  
8             if(!freq[i]) {  
9                 freq[i] = true;  
10                ds.add(nums[i]);  
11                recurPermute(nums, ds, ans, freq);  
12                ds.remove(ds.size() - 1);  
13                freq[i] = false;  
14            }  
15        }  
16    }  
17 }  
18 public List<List<Integer>> permute(int[] nums) {  
19     List<List<Integer>> ans = new ArrayList<>();  
20     List<Integer> ds = new ArrayList<>();  
21     boolean freq[] = new boolean[nums.length];  
22     recurPermute(nums, ds, ans, freq);  
23     return ans;  
24 }  
25 }
```

```
1 class Solution {
2 private:
3     void recurPermute(vector<int> &ds, vector<int> &nums, vector<vector<int>> &ans, int freq[]) {
4         if(ds.size() == nums.size()) {
5             ans.push_back(ds);
6             return;
7         }
8         for(int i = 0; i < nums.size(); i++) {
9             if(!freq[i]) {
10                 ds.push_back(nums[i]);
11                 freq[i] = 1;
12                 recurPermute(ds, nums, ans, freq);
13                 freq[i] = 0;
14                 ds.pop_back();
15             }
16         }
17     }
18 }
19 public:
20     vector<vector<int>> permute(vector<int>& nums) {
21         vector<vector<int>> ans;
22         vector<int> ds;
23         int freq[nums.size()];
24         for(int i = 0; i < nums.size(); i++) freq[i] = 0;
25         recurPermute(ds, nums, ans, freq);
26         return ans;
27     }
28 };
```

$[1, 2, 3]$

$[1, 2, 3]$ $n=3$
 \downarrow \downarrow \downarrow
 1 2 3

$\begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \\ 2 & 1 & 3 \\ 2 & 3 & 1 \\ 3 & 2 & 1 \\ 3 & 1 & 2 \end{bmatrix}$



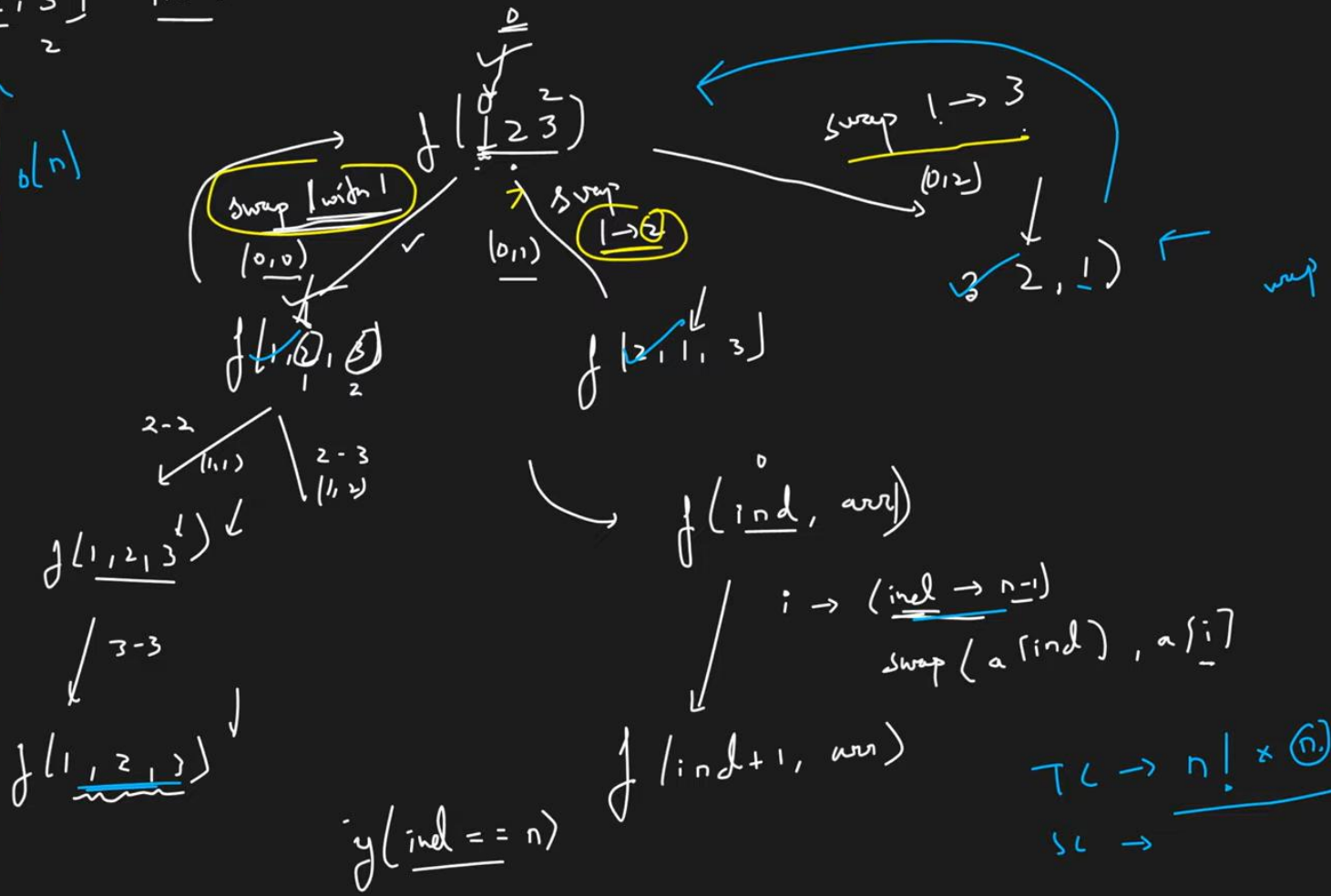
$[1, 2, 3]$ $n=3$

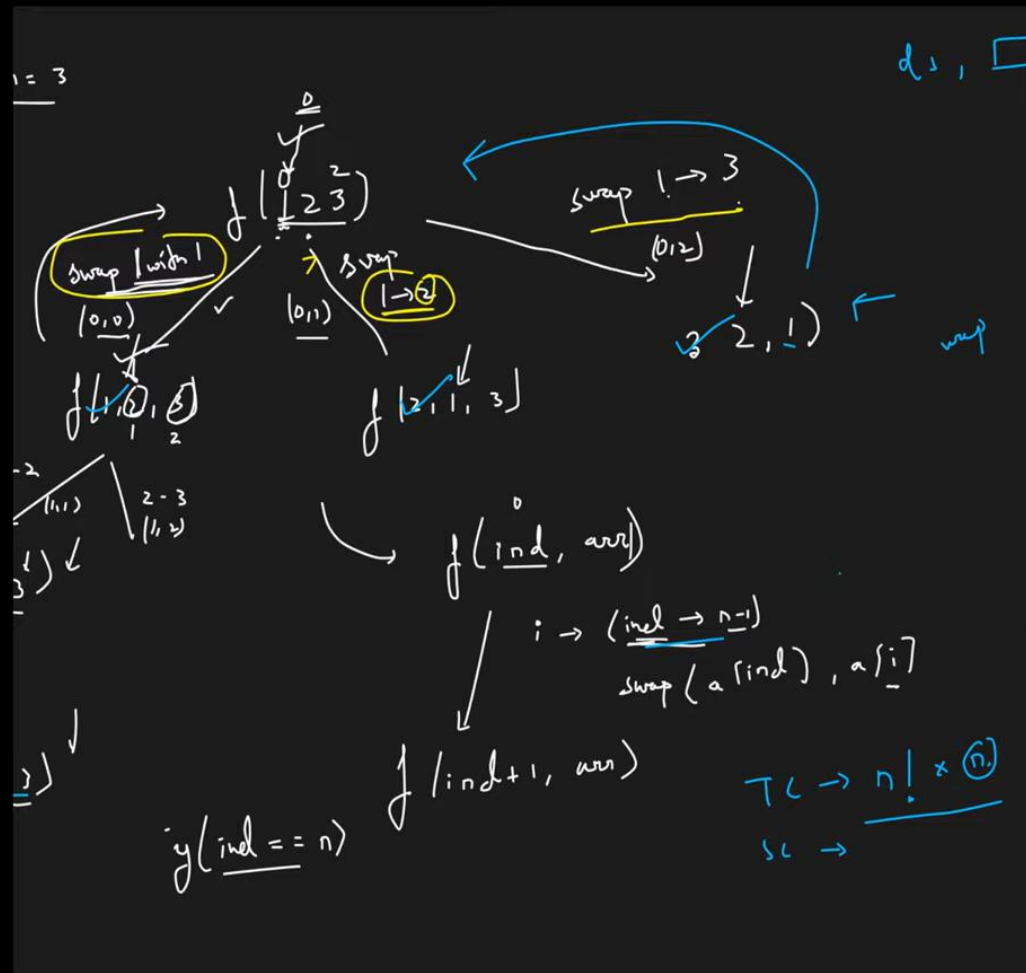
1	2	3
1	3	2
2	1	3
2	3	1
3	2	1
3	1	2

ans.

$O(n!)$

$O(n)$

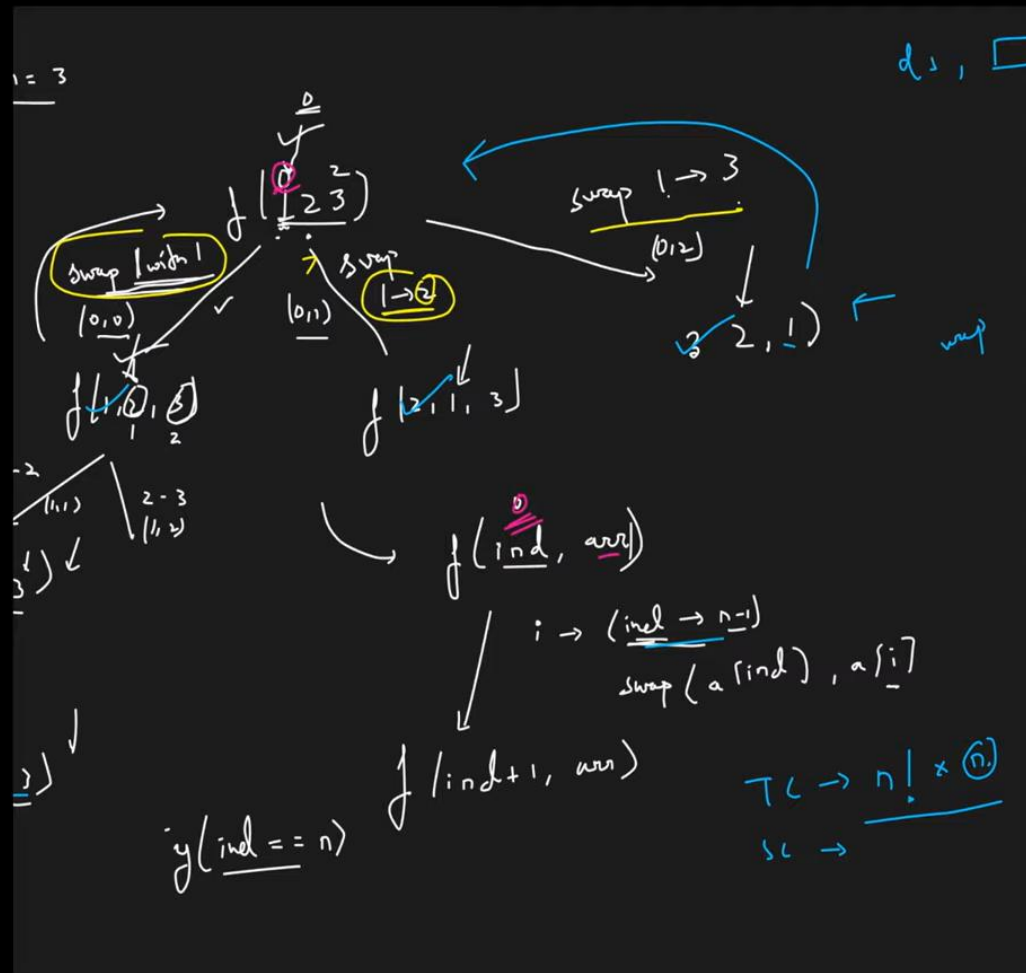




```

1 class Solution {
2     private void recurPermute(int index, int[] nums, List<List<Integer>> ans) {
3         if(index == nums.length) {
4             // copy the ds to ans
5             List<Integer> ds = new ArrayList<>();
6             for(int i = 0; i < nums.length; i++) {
7                 ds.add(nums[i]);
8             }
9             ans.add(new ArrayList<>(ds));
10            return;
11        }
12        for(int i = index; i < nums.length; i++) {
13            swap(i, index, nums);
14            recurPermute(index+1, nums, ans);
15            swap(i, index, nums);
16        }
17    }
18    private void swap(int i, int j, int[] nums) {
19        int t = nums[i];
20        nums[i] = nums[j];
21        nums[j] = t;
22    }
23    public List<List<Integer>> permute(int[] nums) {
24        List<List<Integer>> ans = new ArrayList<>();
25        recurPermute(0, nums, ans);
26        return ans;
27    }
28 }

```



```

1 class Solution {
2 private:
3     void recurPermute(int index, vector<int> &nums, vector<vector<int>> &ans) {
4         if(index == nums.size()) {
5             ans.push_back(nums);
6             return;
7         }
8         for(int i = index; i < nums.size(); i++) {
9             swap(nums[index], nums[i]);
10            recurPermute(index+1, nums, ans);
11            swap(nums[index], nums[i]);
12        }
13    }
14 public:
15     vector<vector<int>> permute(vector<int> &nums) {
16         vector<vector<int>> ans;
17         recurPermute(0, nums, ans);
18         return ans;
19     }
20 };
21

```