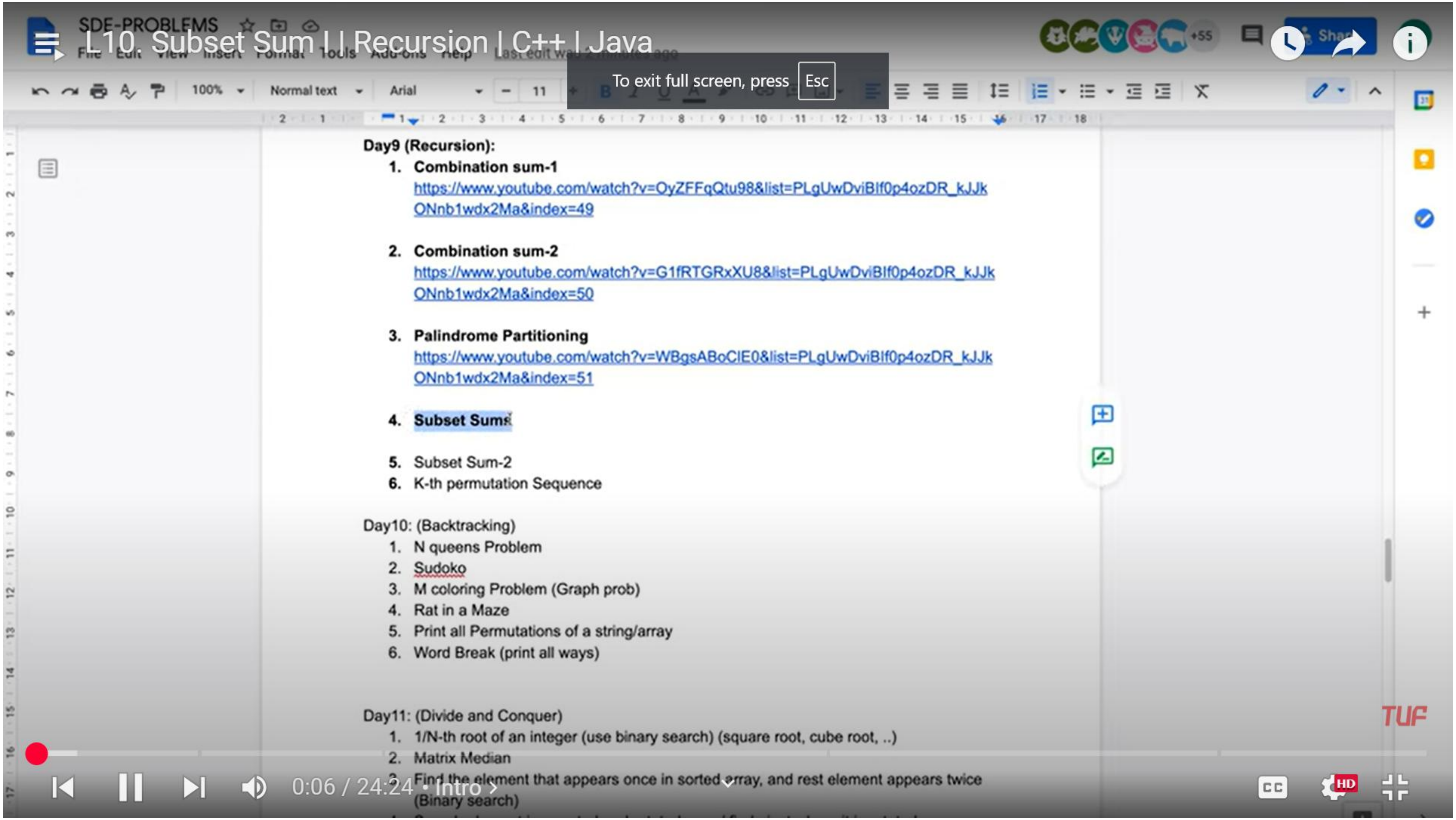

SDE-PROBLEMS
L10. Subset Sum I | Recursion | C++ | Java
File Edit View Insert Format Tools Add-ons Help
To exit full screen, press Esc
100%
Normal text
Arial
11
+55
Share
TUF
0:06 / 24:24 • Intro ›
HD

Day9 (Recursion):

1. **Combination sum-1**
   https://www.youtube.com/watch?v=OyZFFqQtu98&list=PLgUwDviBIf0p4ozDR_kJJkONnb1wdx2Ma&index=49

2. **Combination sum-2**
   https://www.youtube.com/watch?v=G1fRTGRxXU8&list=PLgUwDviBIf0p4ozDR_kJJkONnb1wdx2Ma&index=50

3. **Palindrome Partitioning**
   https://www.youtube.com/watch?v=WBgsABoClE0&list=PLgUwDviBIf0p4ozDR_kJJkONnb1wdx2Ma&index=51

4. **Subset Sums**

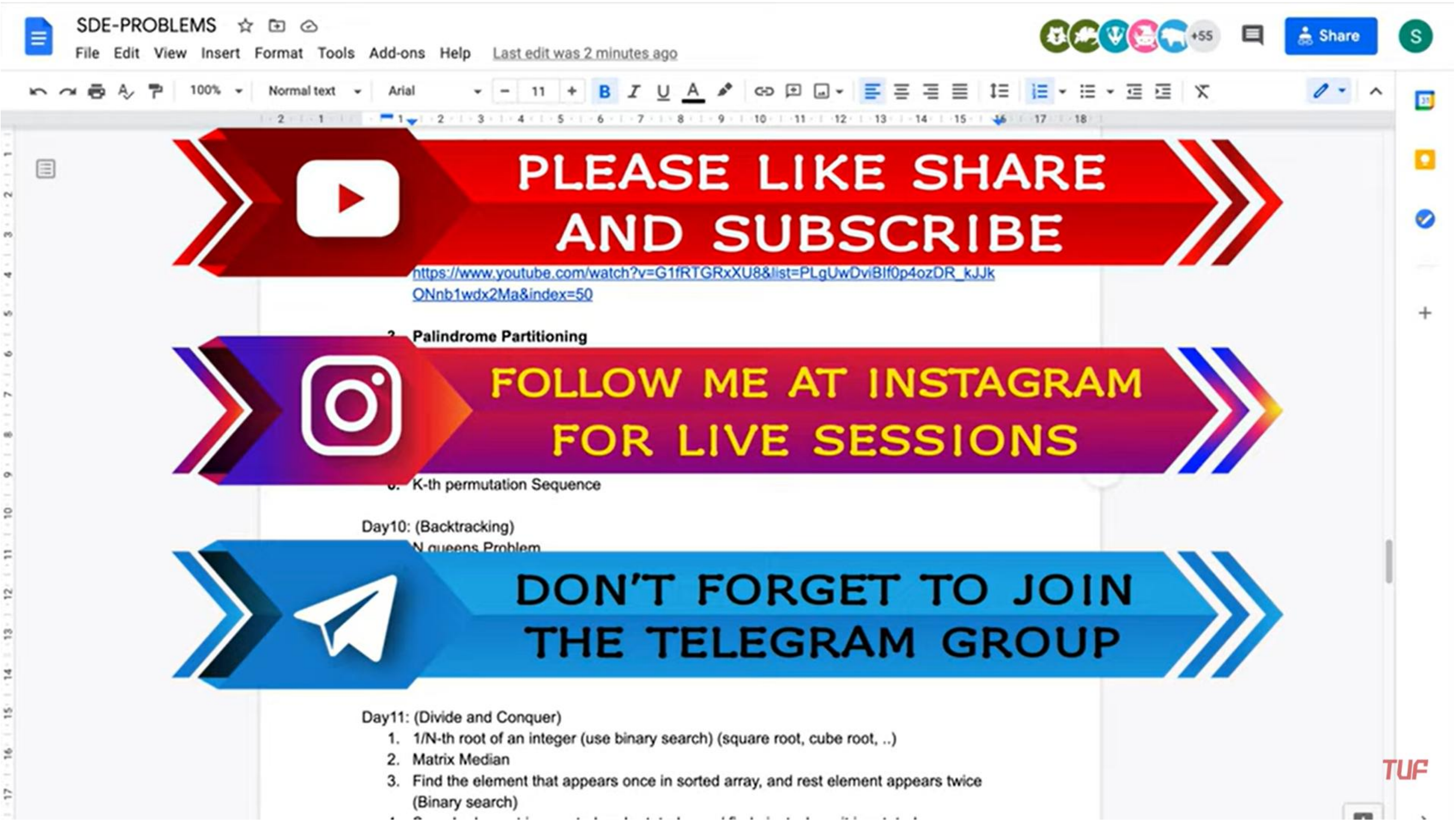
5. Subset Sum-2
6. K-th permutation Sequence

Day10: (Backtracking)

1. N queens Problem
2. Sudoko
3. M coloring Problem (Graph prob)
4. Rat in a Maze
5. Print all Permutations of a string/array
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)


SDE-PROBLEMS
File Edit View Insert Format Tools Add-ons Help
Last edit was 2 minutes ago
Share
Normal text
Arial
PLEASE LIKE SHARE AND SUBSCRIBE
FOLLOW ME AT INSTAGRAM FOR LIVE SESSIONS
DON'T FORGET TO JOIN THE TELEGRAM GROUP
TUF

https://www.youtube.com/watch?v=G1fRTGRxXU8&list=PLgUwDviBIf0p4ozDR_kJJkONnb1wdx2Ma&index=50

2. **Palindrome Partitioning**

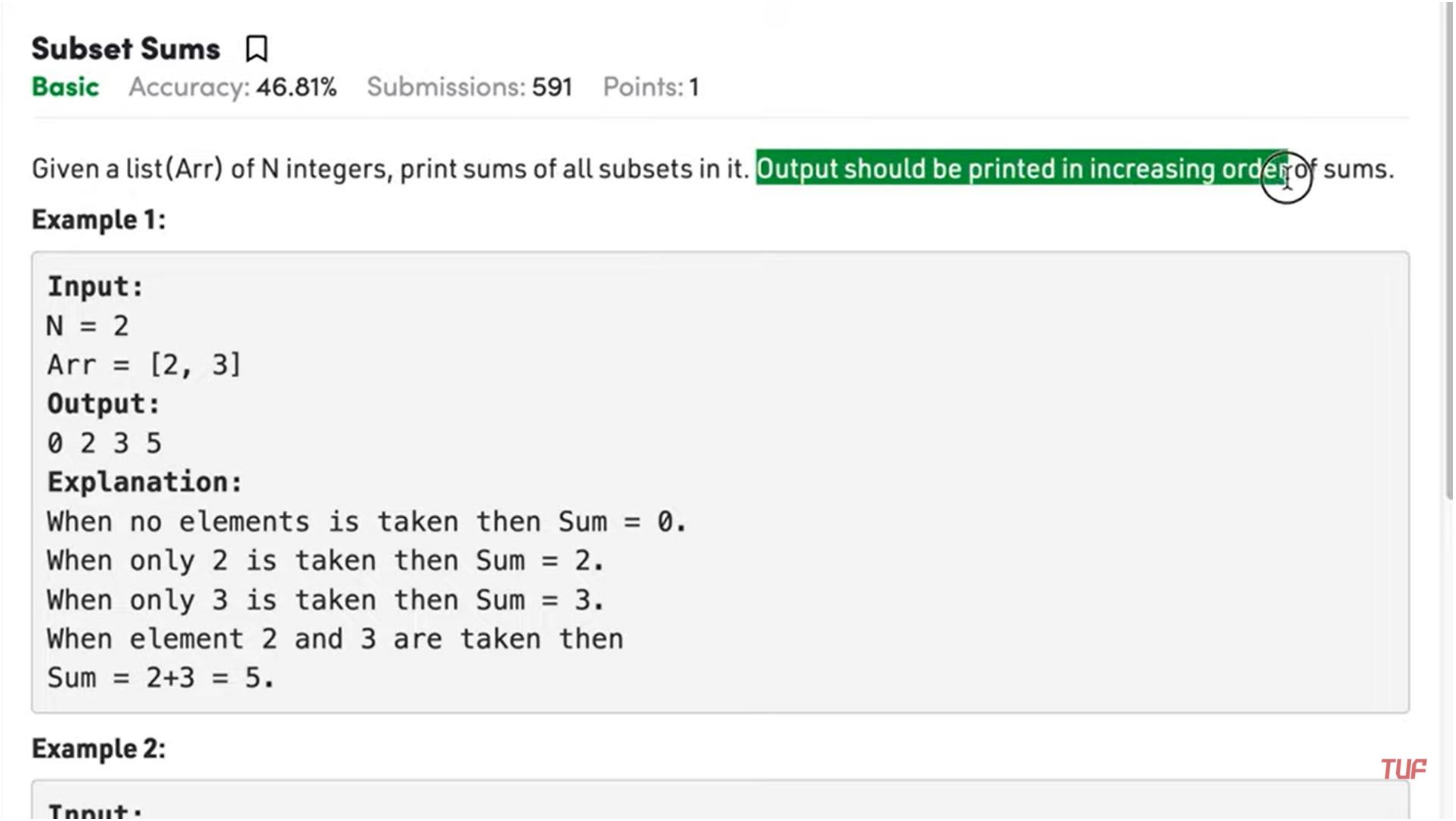
6. K-th permutation Sequence

Day10: (Backtracking)

N queens Problem

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)



Subset Sums

Basic Accuracy: 46.81% Submissions: 591 Points: 1

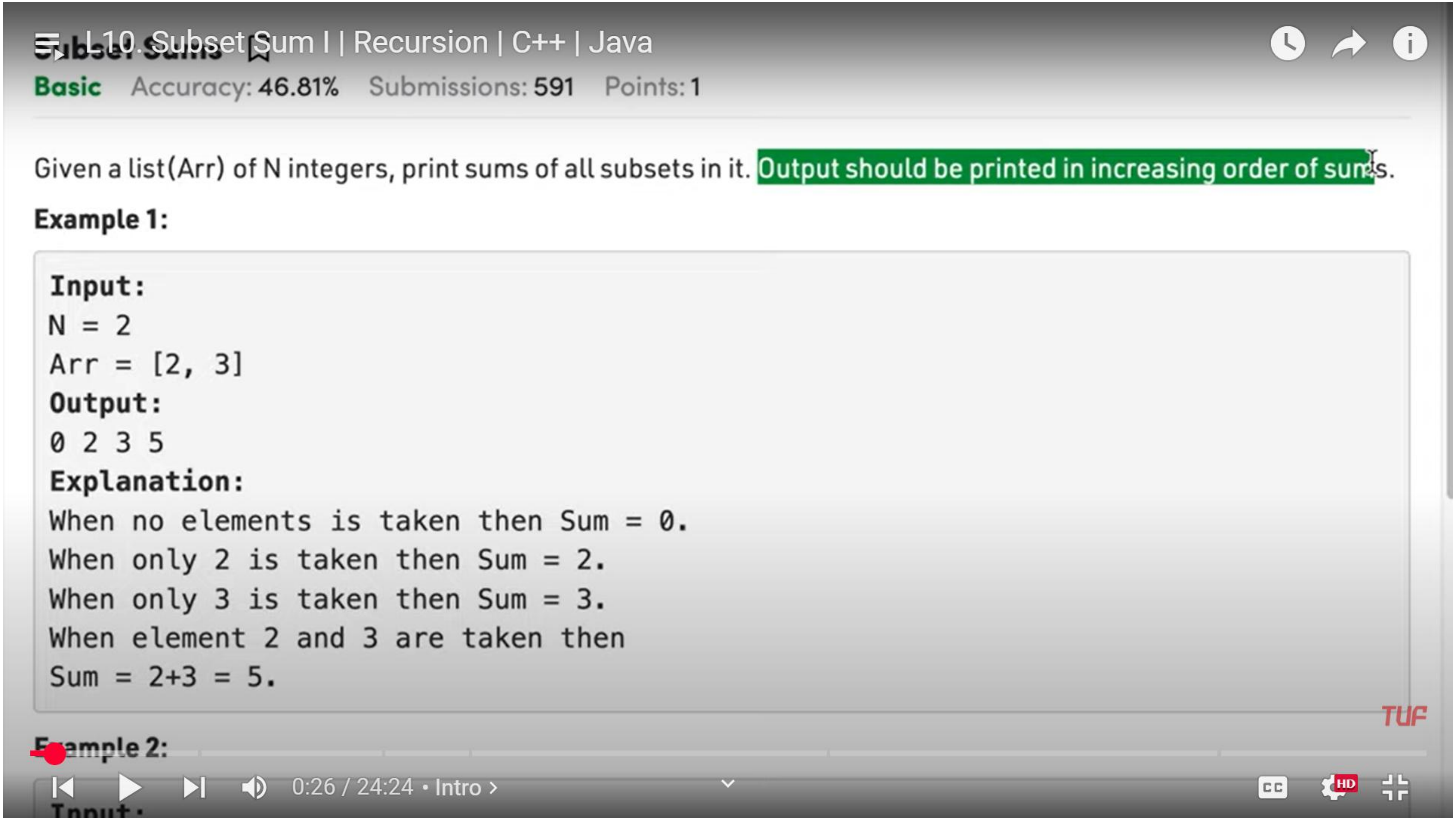
Given a list(Arr) of N integers, print sums of all subsets in it. Output should be printed in increasing order of sums.

Example 1:

```
Input:
N = 2
Arr = [2, 3]
Output:
0 2 3 5
Explanation:
When no elements is taken then Sum = 0.
When only 2 is taken then Sum = 2.
When only 3 is taken then Sum = 3.
When element 2 and 3 are taken then
Sum = 2+3 = 5.
```

Example 2:

```
Input:
```



Subset Sums

Basic Accuracy: 46.81% Submissions: 591 Points: 1

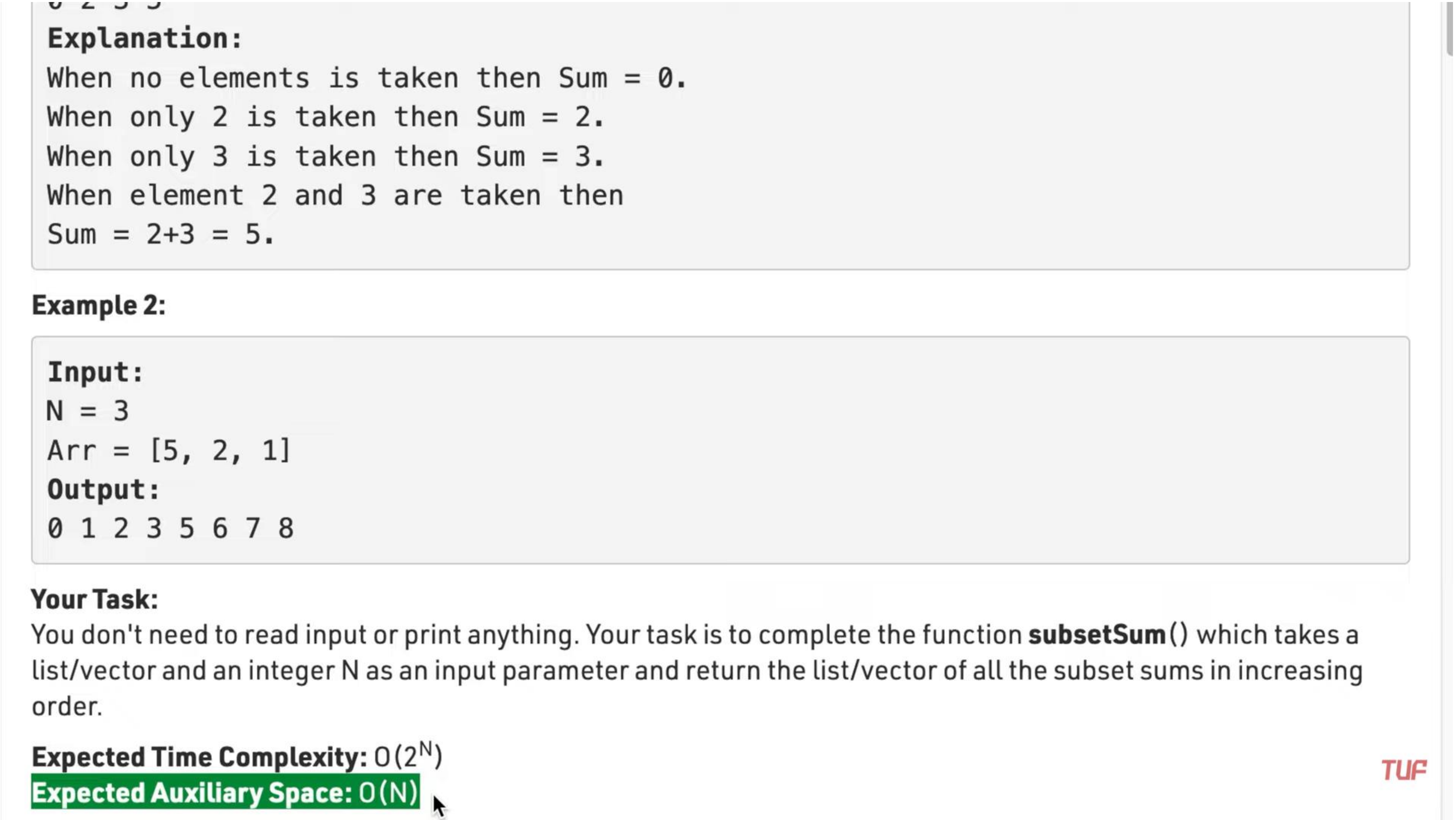
Given a list(Arr) of N integers, print sums of all subsets in it. Output should be printed in increasing order of sums.

Example 1:

```
Input:
N = 2
Arr = [2, 3]
Output:
0 2 3 5
Explanation:
When no elements is taken then Sum = 0.
When only 2 is taken then Sum = 2.
When only 3 is taken then Sum = 3.
When element 2 and 3 are taken then
Sum = 2+3 = 5.
```

Example 2:

```
Input:
```



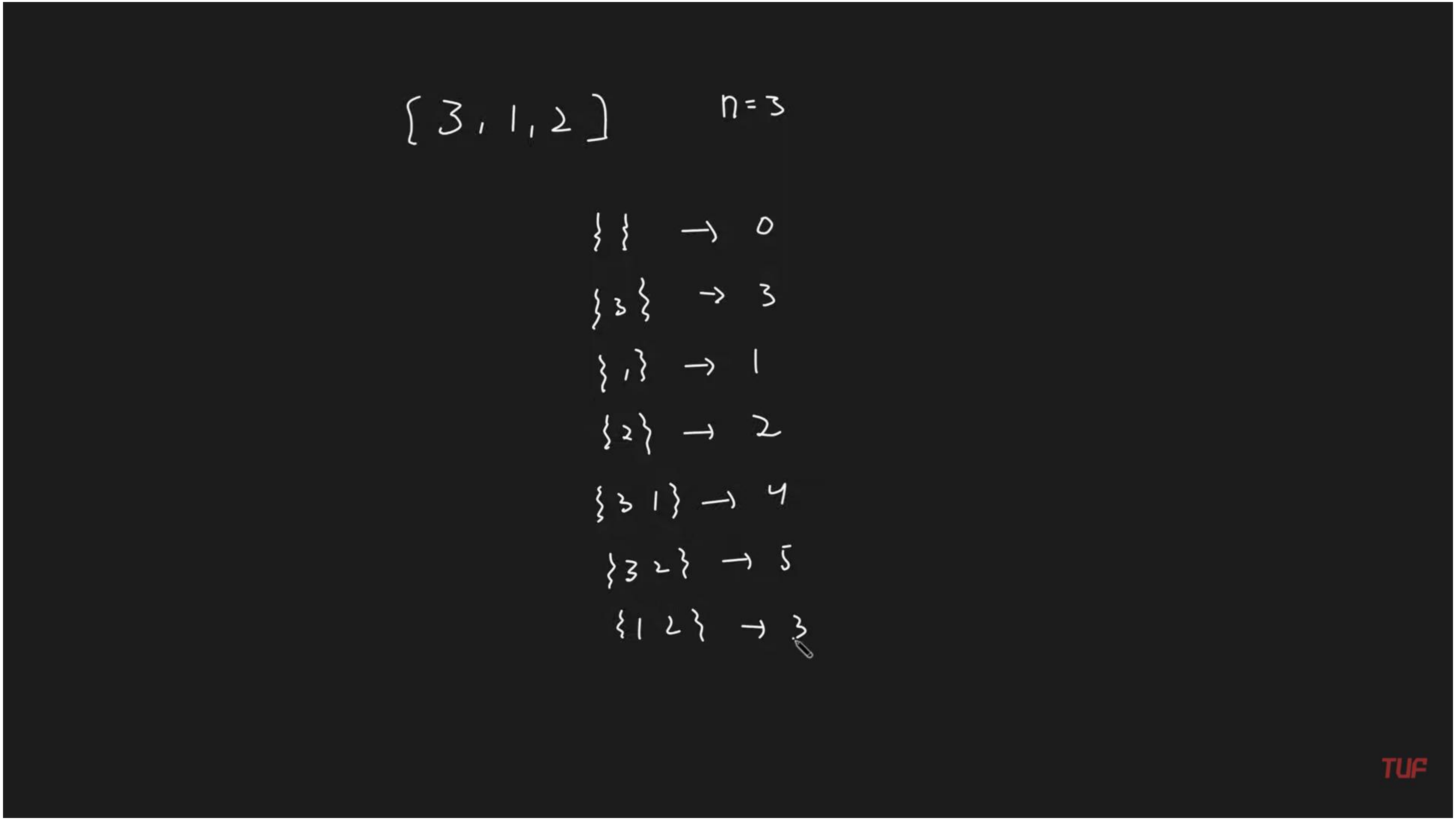
```
0 2 3 5
Explanation:
When no elements is taken then Sum = 0.
When only 2 is taken then Sum = 2.
When only 3 is taken then Sum = 3.
When element 2 and 3 are taken then
Sum = 2+3 = 5.
```

Example 2:

```
Input:
N = 3
Arr = [5, 2, 1]
Output:
0 1 2 3 5 6 7 8
```

Your Task:
You don't need to read input or print anything. Your task is to complete the function **subsetSum**() which takes a list/vector and an integer N as an input parameter and return the list/vector of all the subset sums in increasing order.

Expected Time Complexity: O(2^N)
Expected Auxiliary Space: O(N)



[3, 1, 2]   n = 3

{ } → 0

{3} → 3

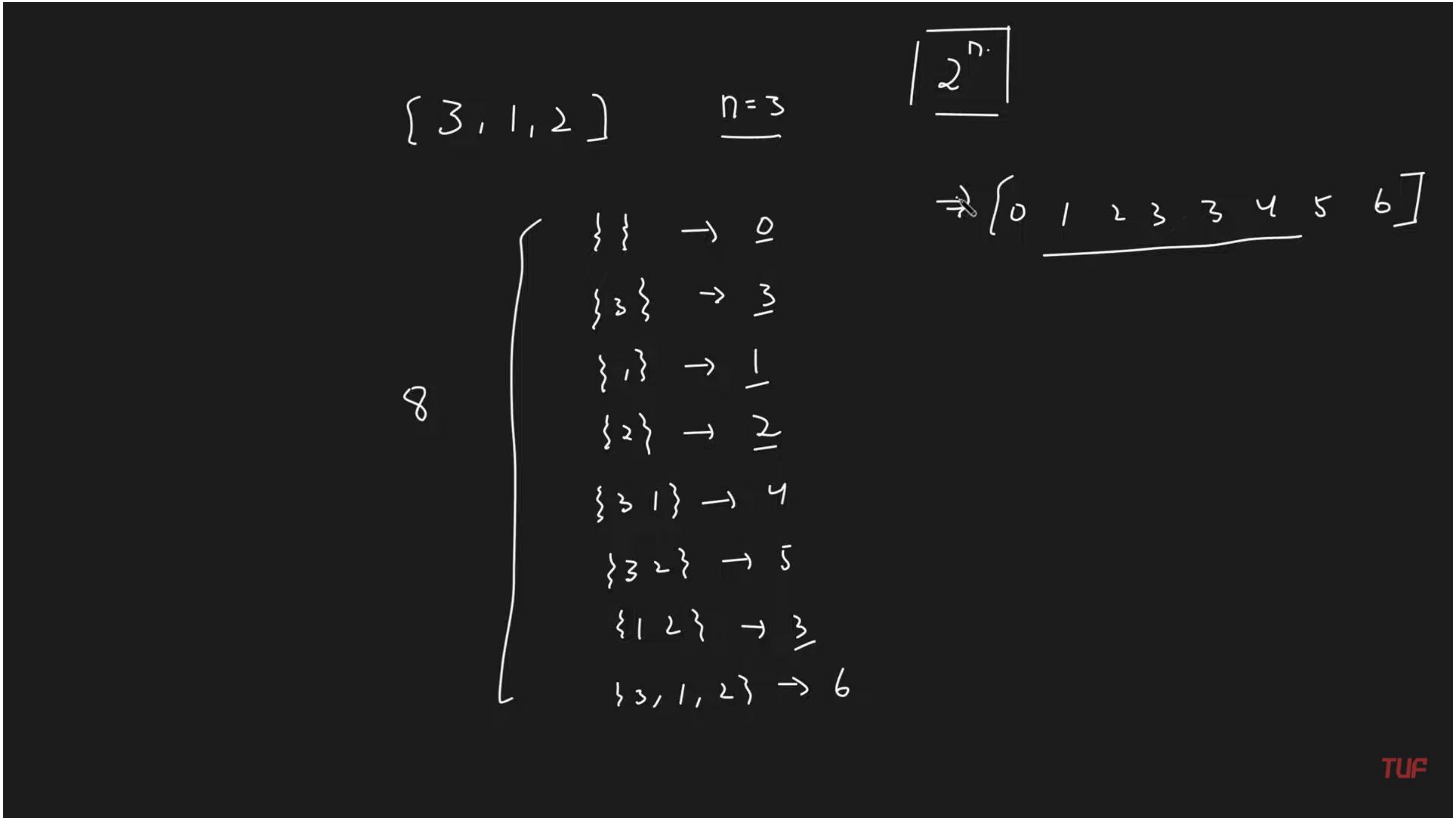
{1} → 1

{2} → 2

{3 1} → 4

{3 2} → 5

{1 2} → 3



[3, 1, 2] $\underline{n=3}$

$\boxed{2^n}$

$\Rightarrow$ [0 1 2 3 3 4 5 6]

8

$\{\,\} \rightarrow \underline{0}$

$\{3\} \rightarrow \underline{3}$

$\{1\} \rightarrow \underline{1}$

$\{2\} \rightarrow \underline{2}$

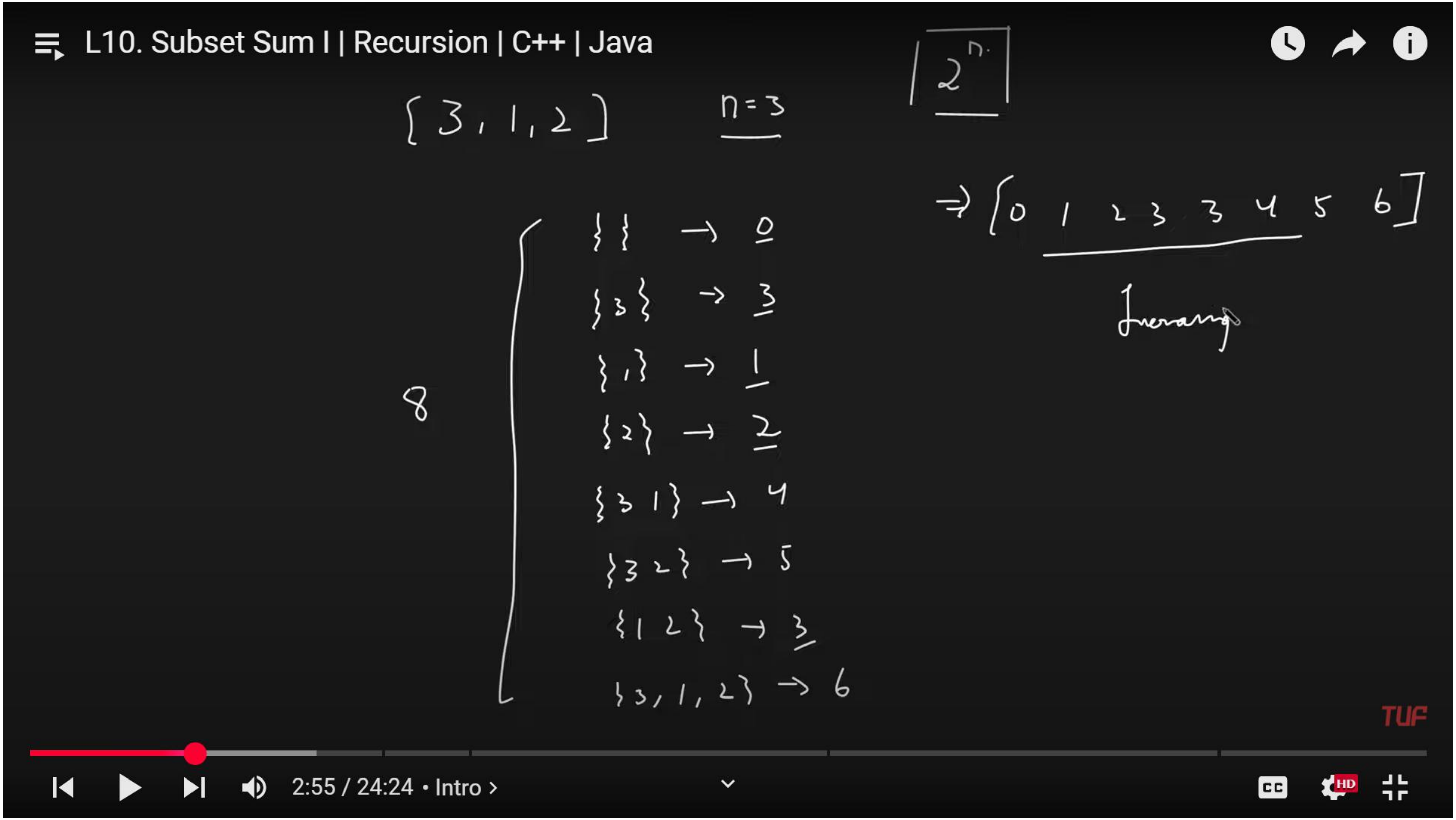
$\{3\ 1\} \rightarrow 4$

$\{3\ 2\} \rightarrow 5$

$\{1\ 2\} \rightarrow \underline{3}$

$\{3, 1, 2\} \rightarrow 6$

[3 1 4]



L10. Subset Sum I | Recursion | C++ | Java

$[3, 1, 2]$ $\underline{n=3}$

$\boxed{2^n}$

8

$\{\} \to \underline{0}$

$\{3\} \to \underline{3}$

$\{1\} \to \underline{1}$

$\{2\} \to \underline{2}$

$\{3\ 1\} \to 4$

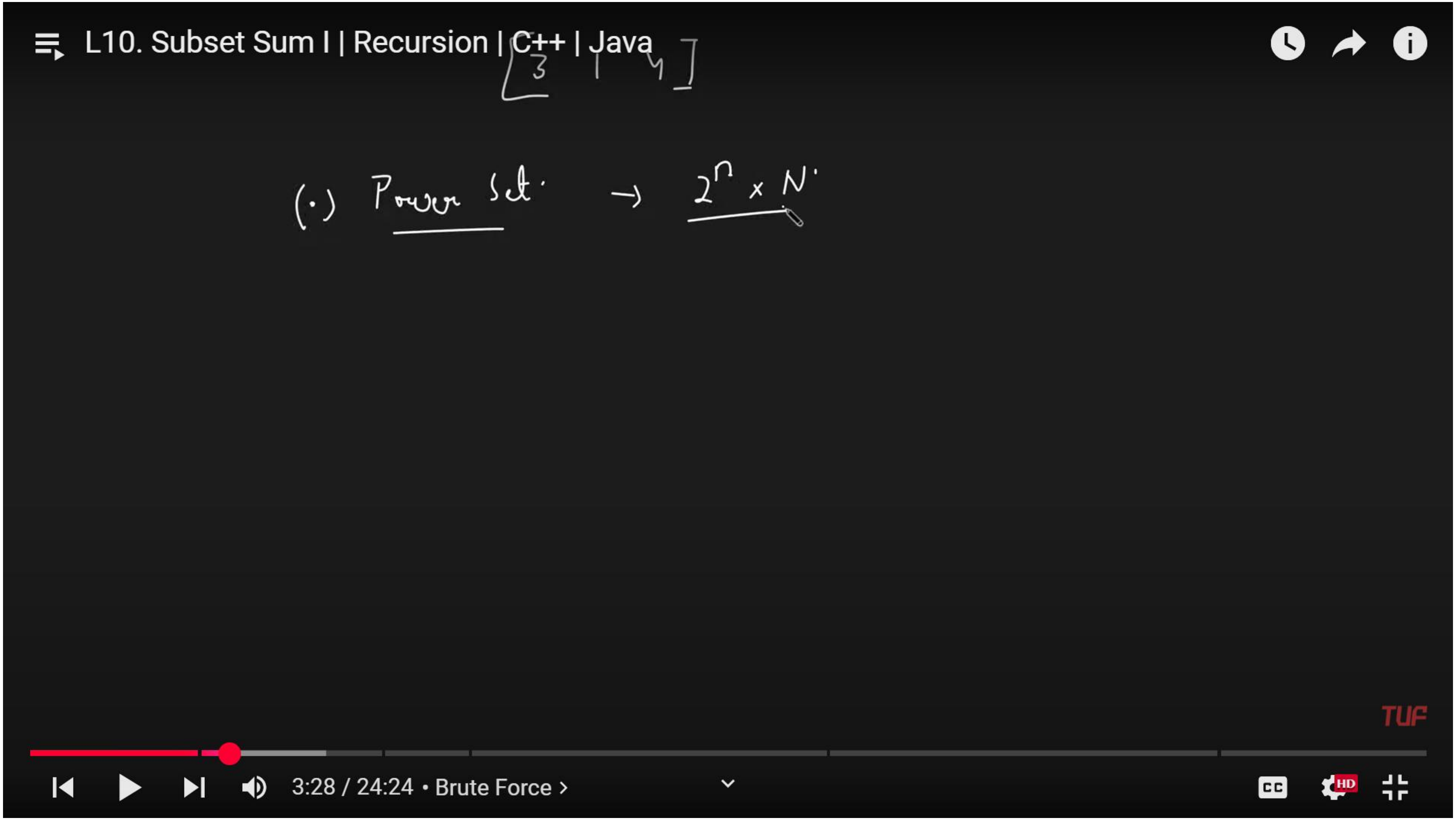
$\{3\ 2\} \to 5$

$\{1\ 2\} \to \underline{3}$

$\{3, 1, 2\} \to 6$

$\Rightarrow [0\ \underline{1\ 2\ 3\ 3\ 4}\ 5\ 6]$

Increasing

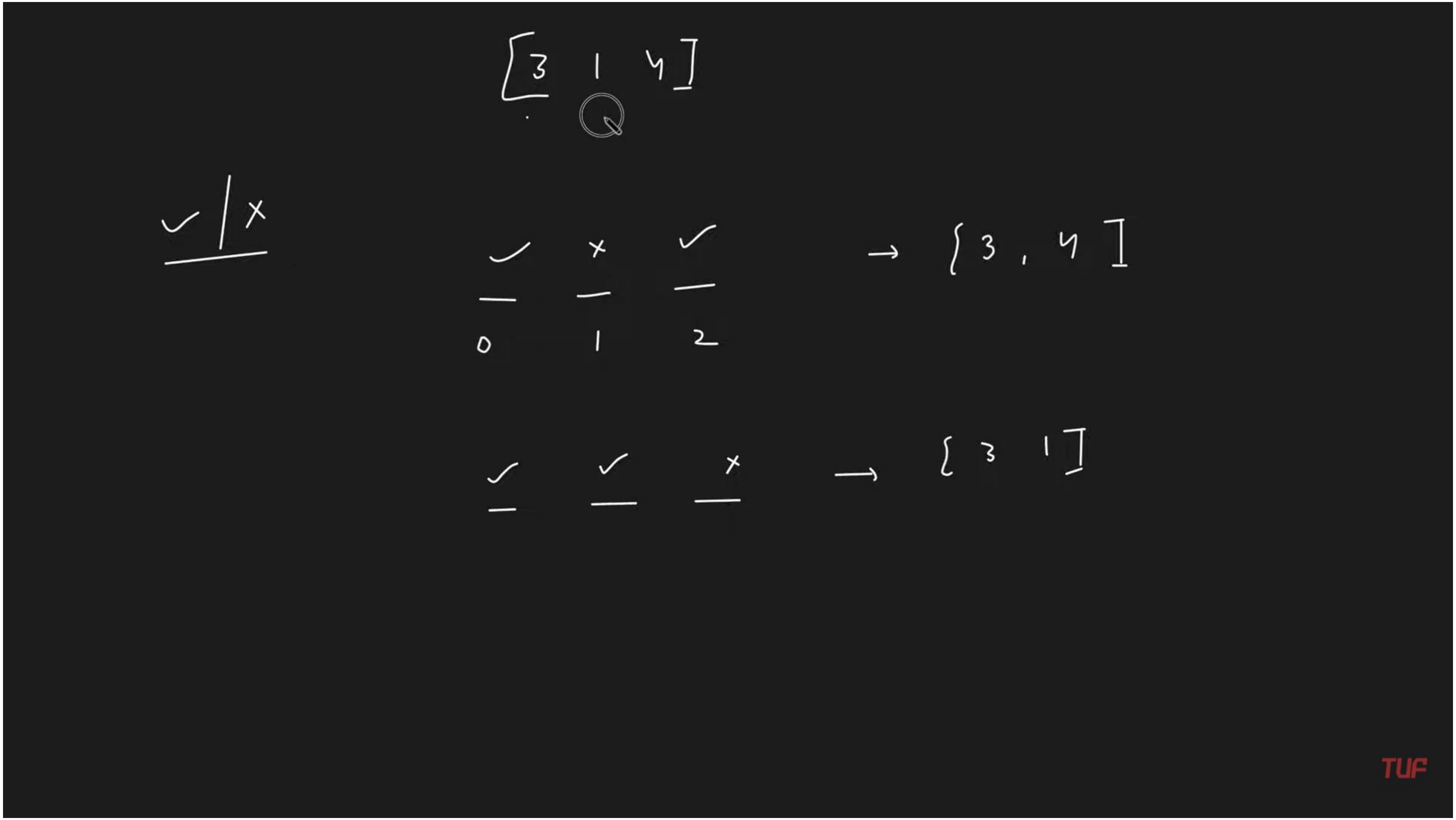


L10. Subset Sum I | Recursion | C++ | Java

$[3 \quad 1 \quad 4]$

(·) Power Set. → $\underline{2^n \times N}$

3:28 / 24:24 • Brute Force ›



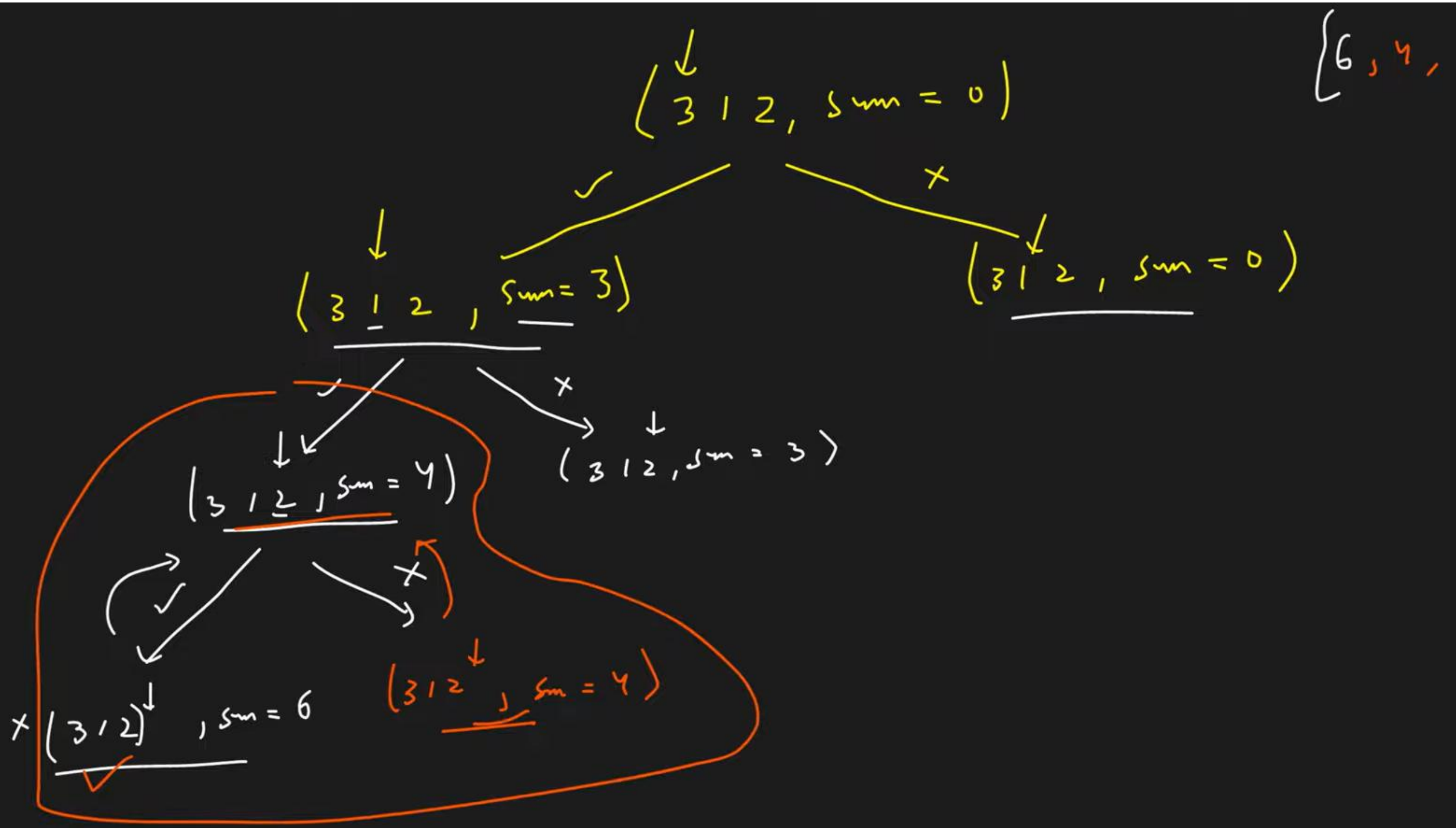
[3 1 4]

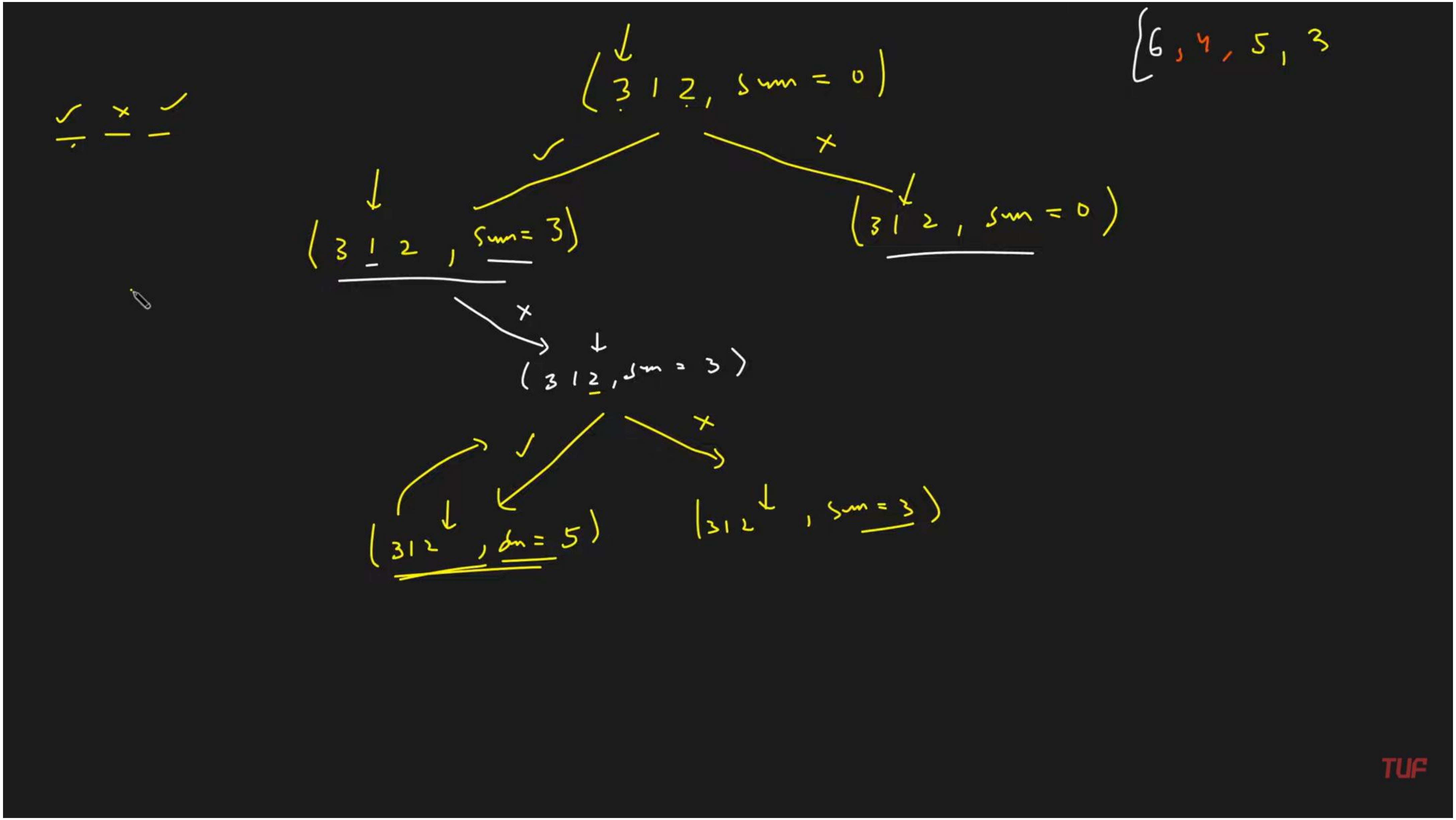
✓/X

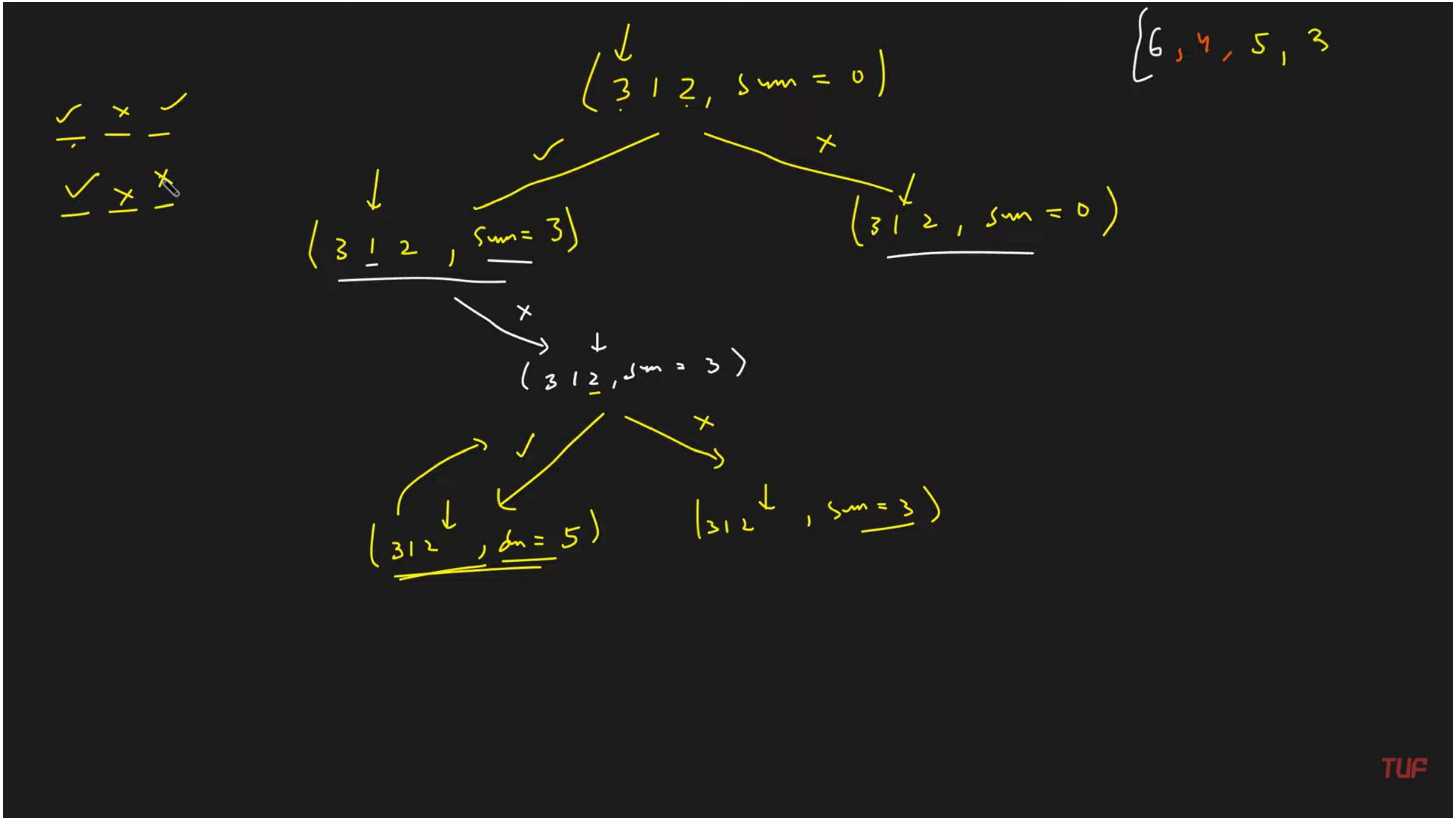
✓ X ✓ → [3, 4]

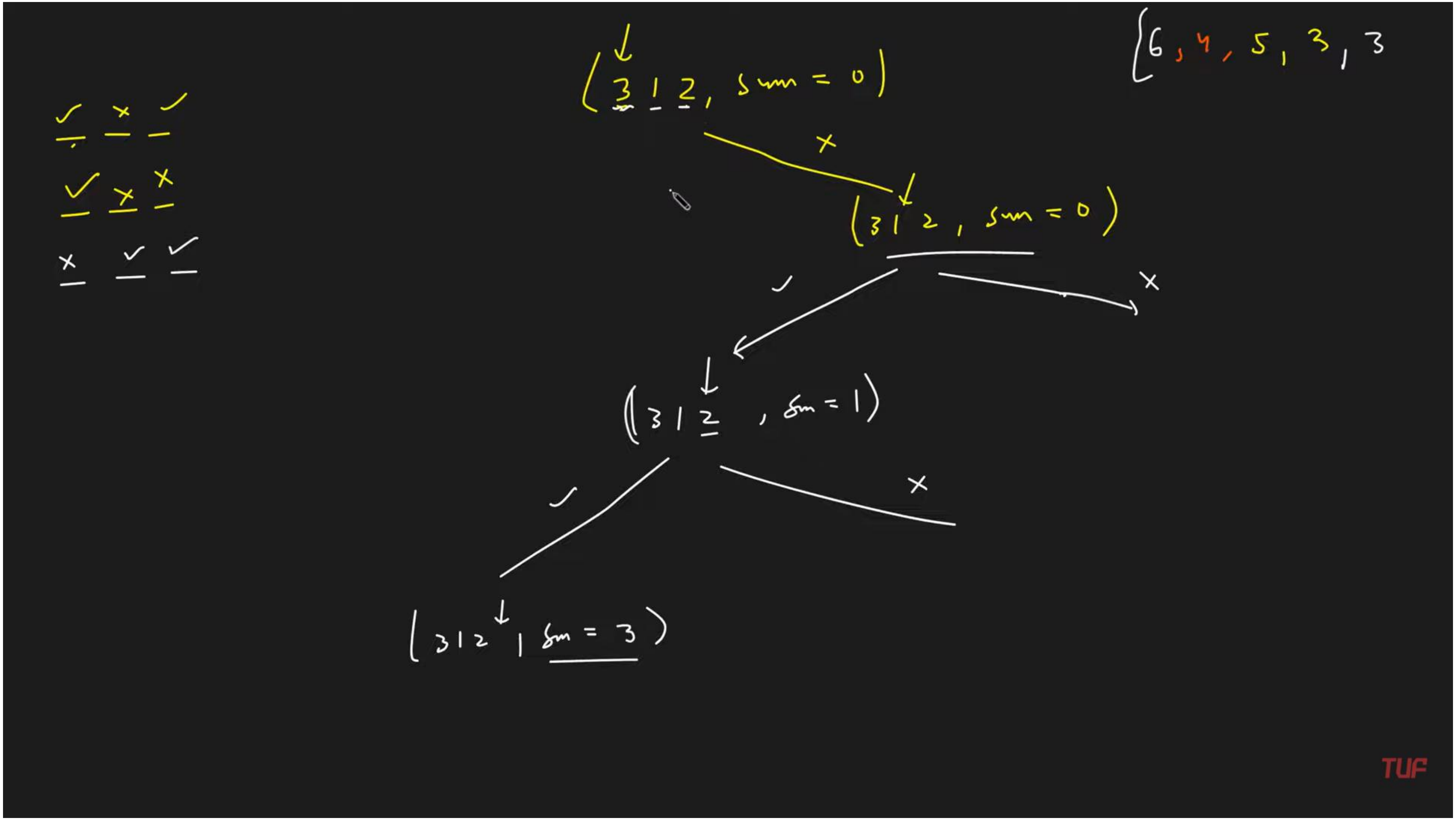
0 1 2

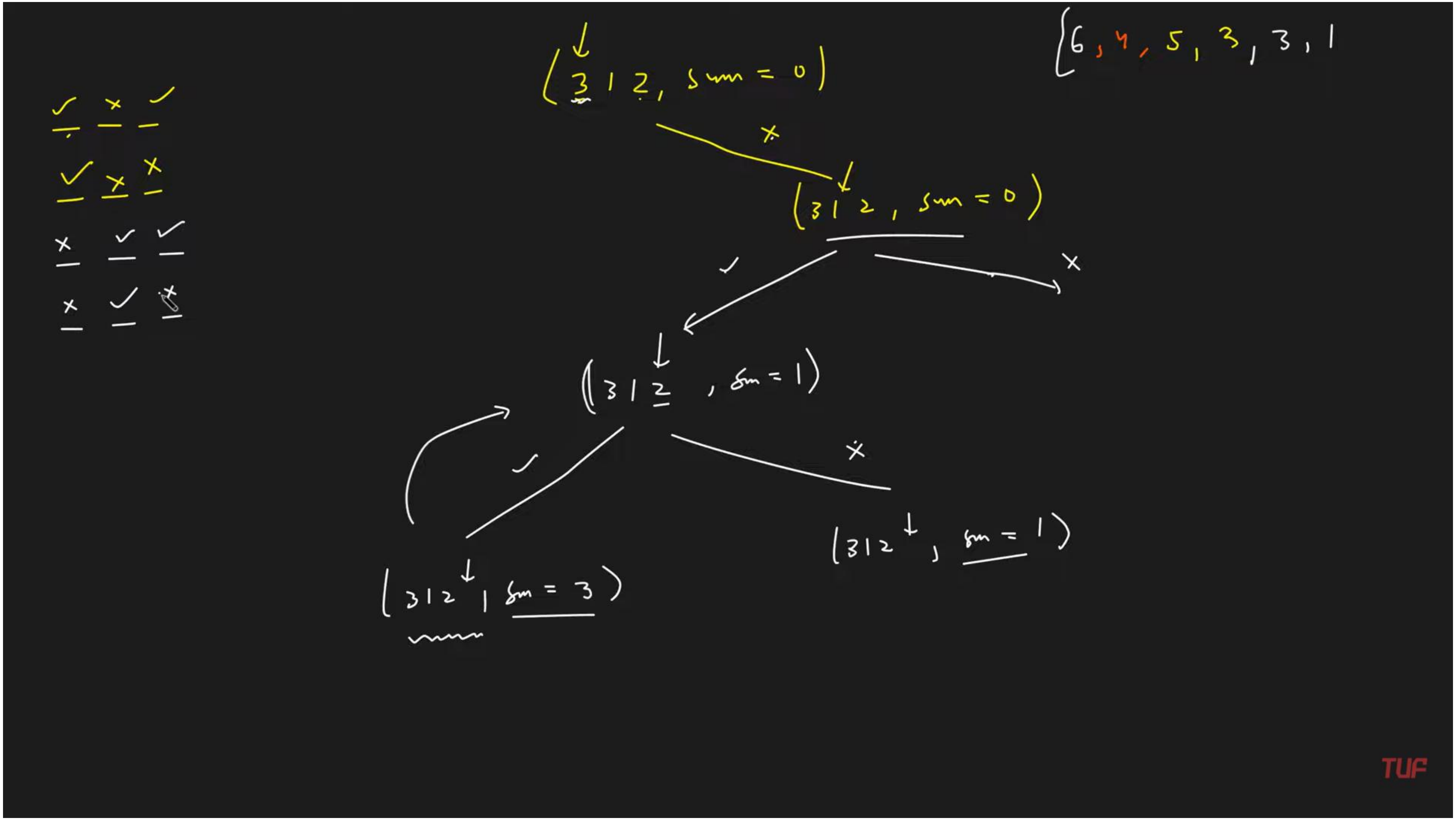
✓ ✓ X → [3 1]

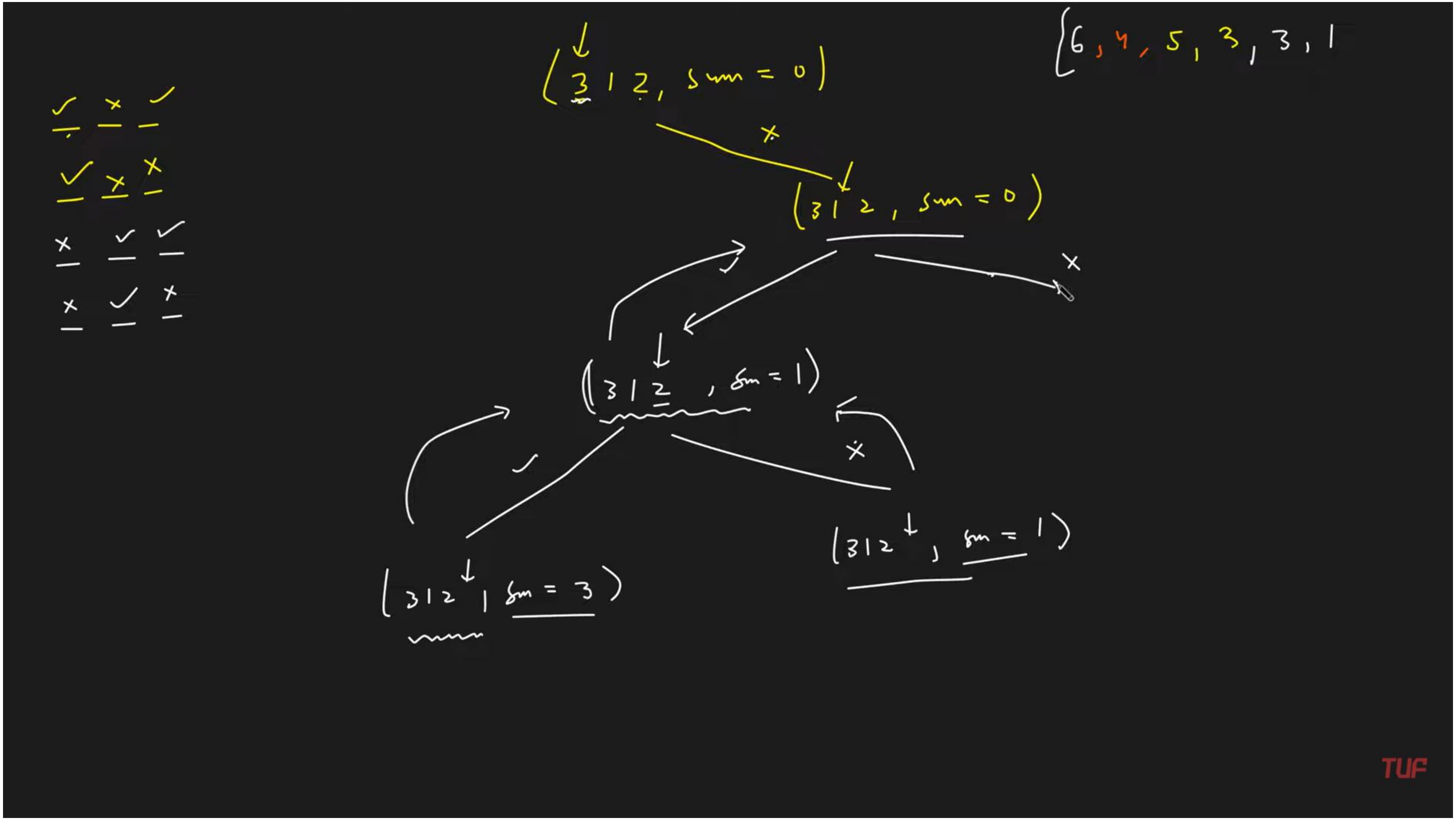

(3 1 2, sum = 0)
✓
✗
(3 1 2, sum = 3)
(3 1 2, sum = 0)
✓
✗
(3 1 2, sum = 4)
(3 1 2, sum = 3)
✓
✗
✗ (3 1 2), sum = 6
✓
(3 1 2, sum = 4)
{6, 4,

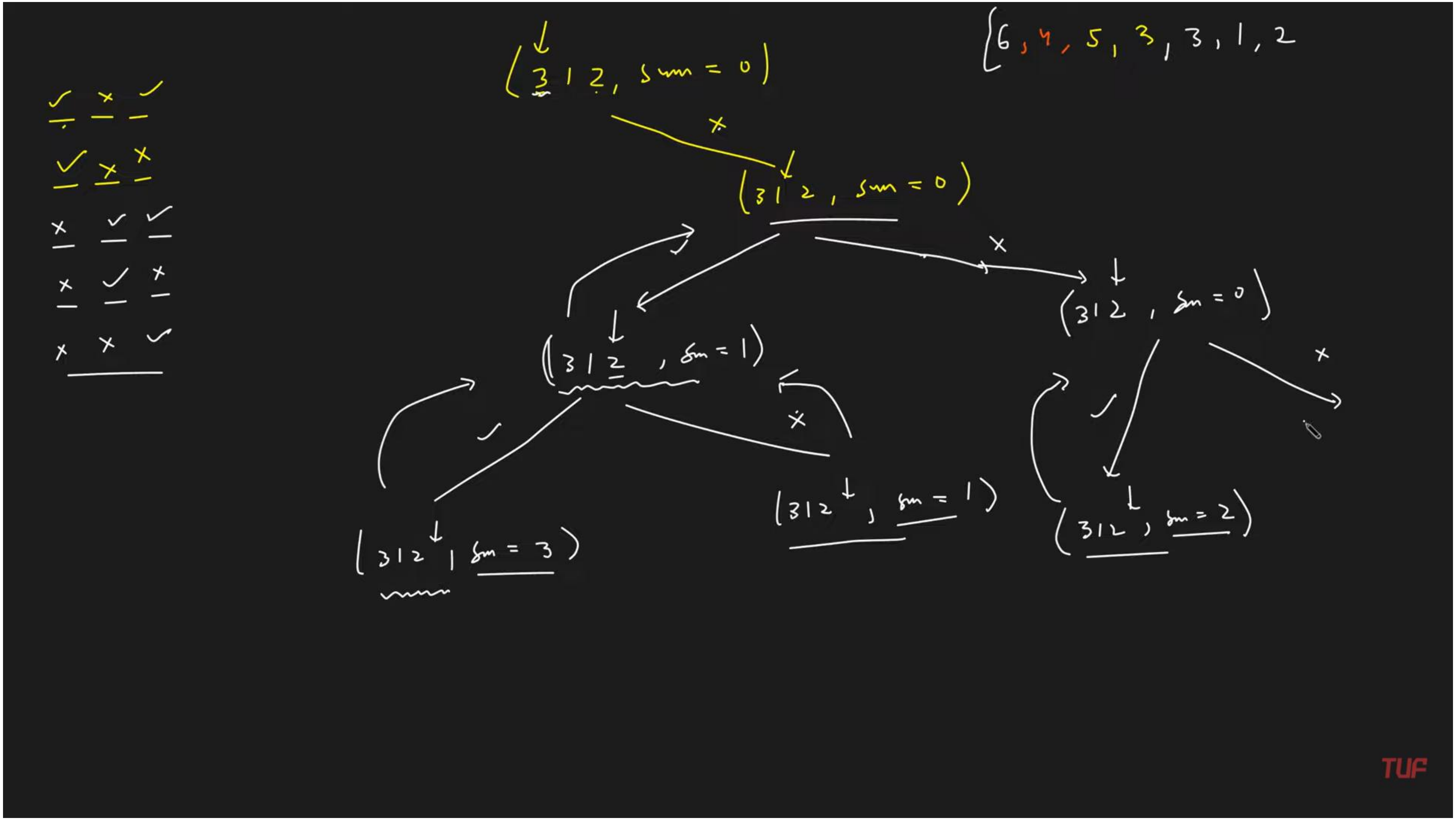

✓ × ✓
(3 1 2, sum = 0)
{6, 4, 5, 3
✓
×
(3 1 2 , sum= 3)
(3 1 2 , sum = 0)
×
(3 1 2, sum = 3)
✓
×
(3 1 2 , sum = 5)
(3 1 2 , sum = 3)

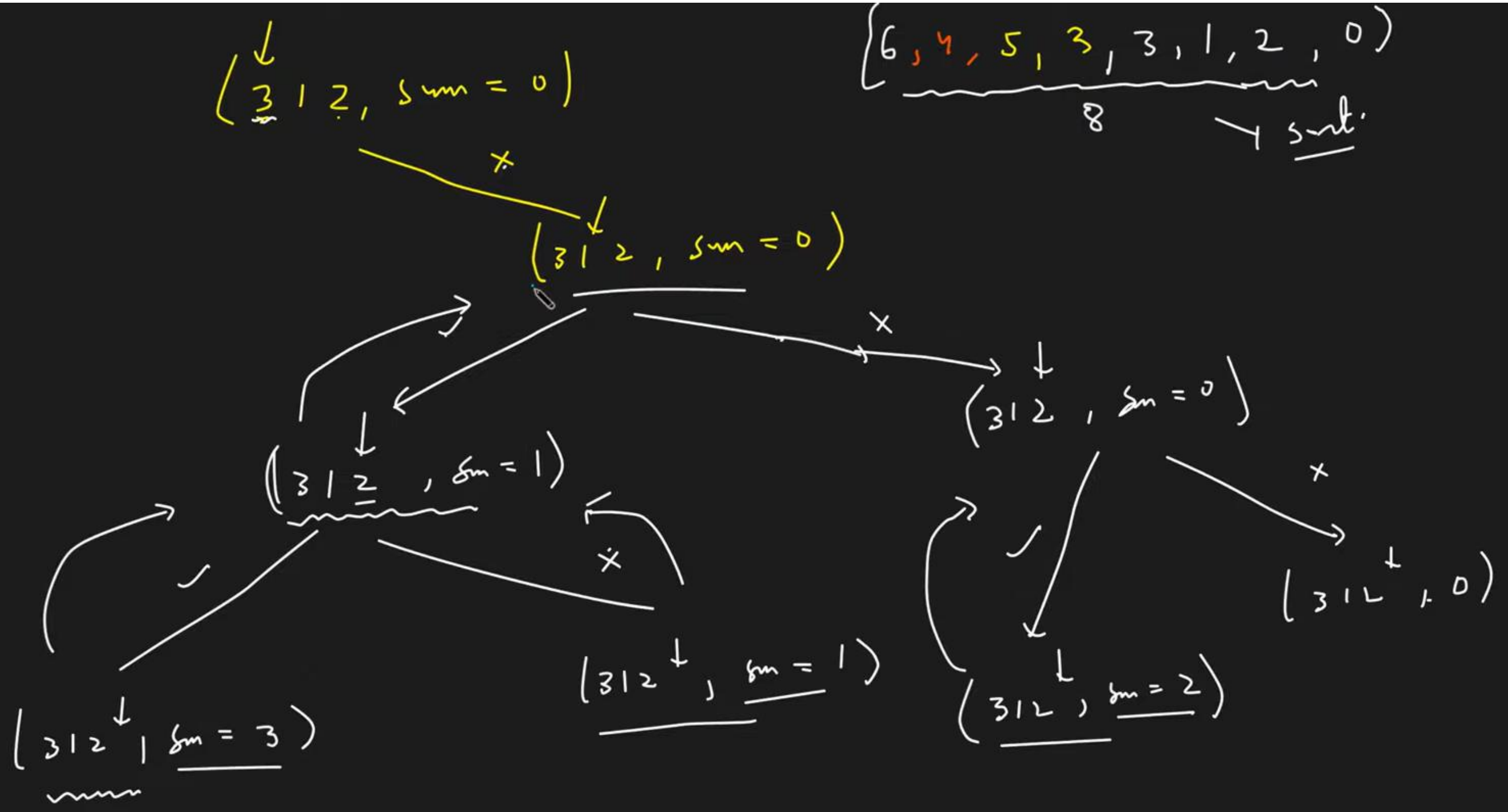

✓ × ✓
✓ × ×
(3 1 2, sum = 0)
{6, 4, 5, 3
✓
×
(3 1 2, sum = 3)
(3 1 2, sum = 0)
×
(3 1 2, sum = 3)
✓
×
(3 1 2, sum = 5)
(3 1 2, sum = 3)
TUF

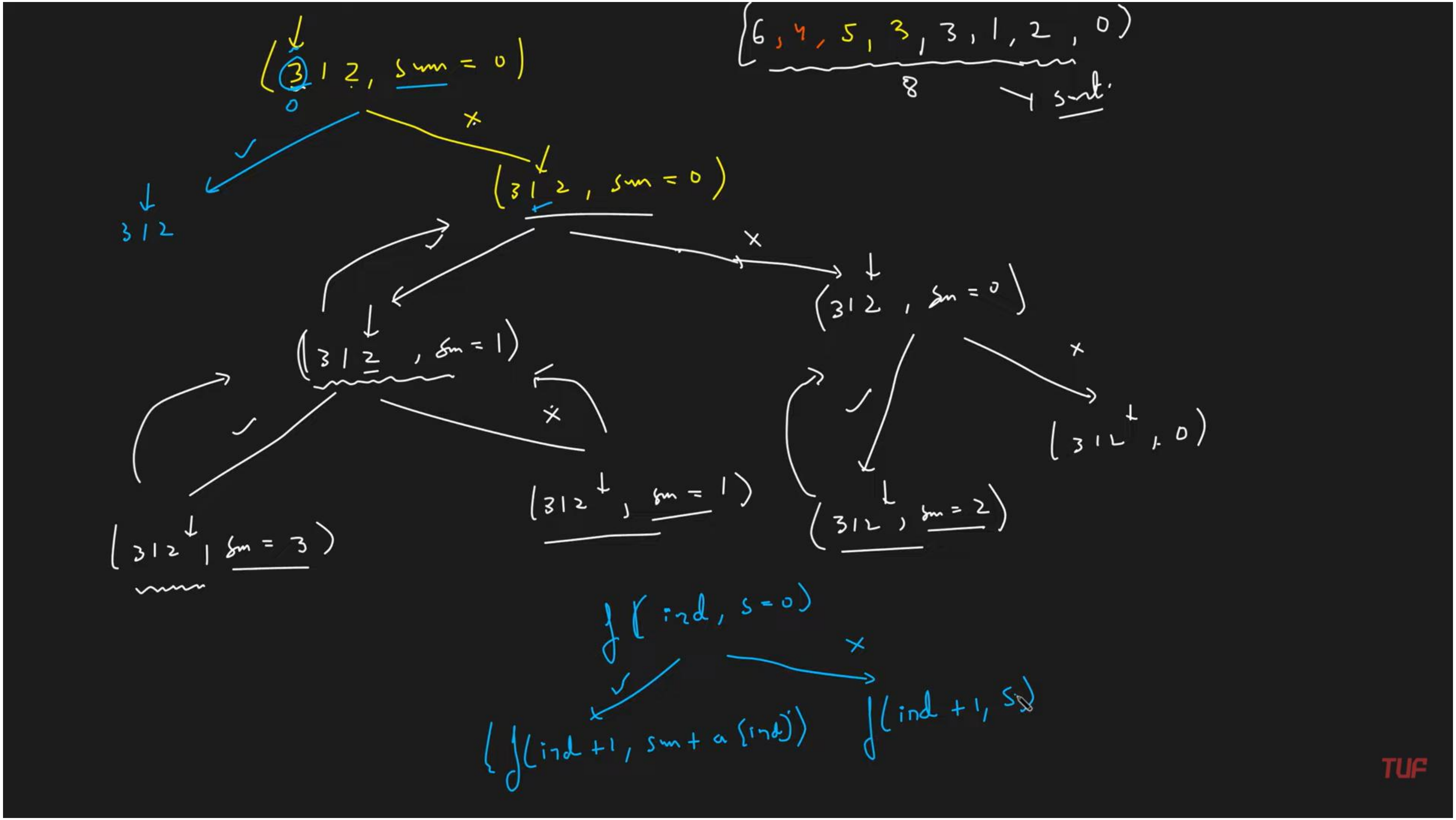

{6, 4, 5, 3, 3
✓ × ✓
✓ × ×
× ✓ ✓
(3 1 2, sum = 0)
×
(3 1 2, sum = 0)
✓
×
(3 1 2, sum = 1)
✓
×
(3 1 2, sum = 3)

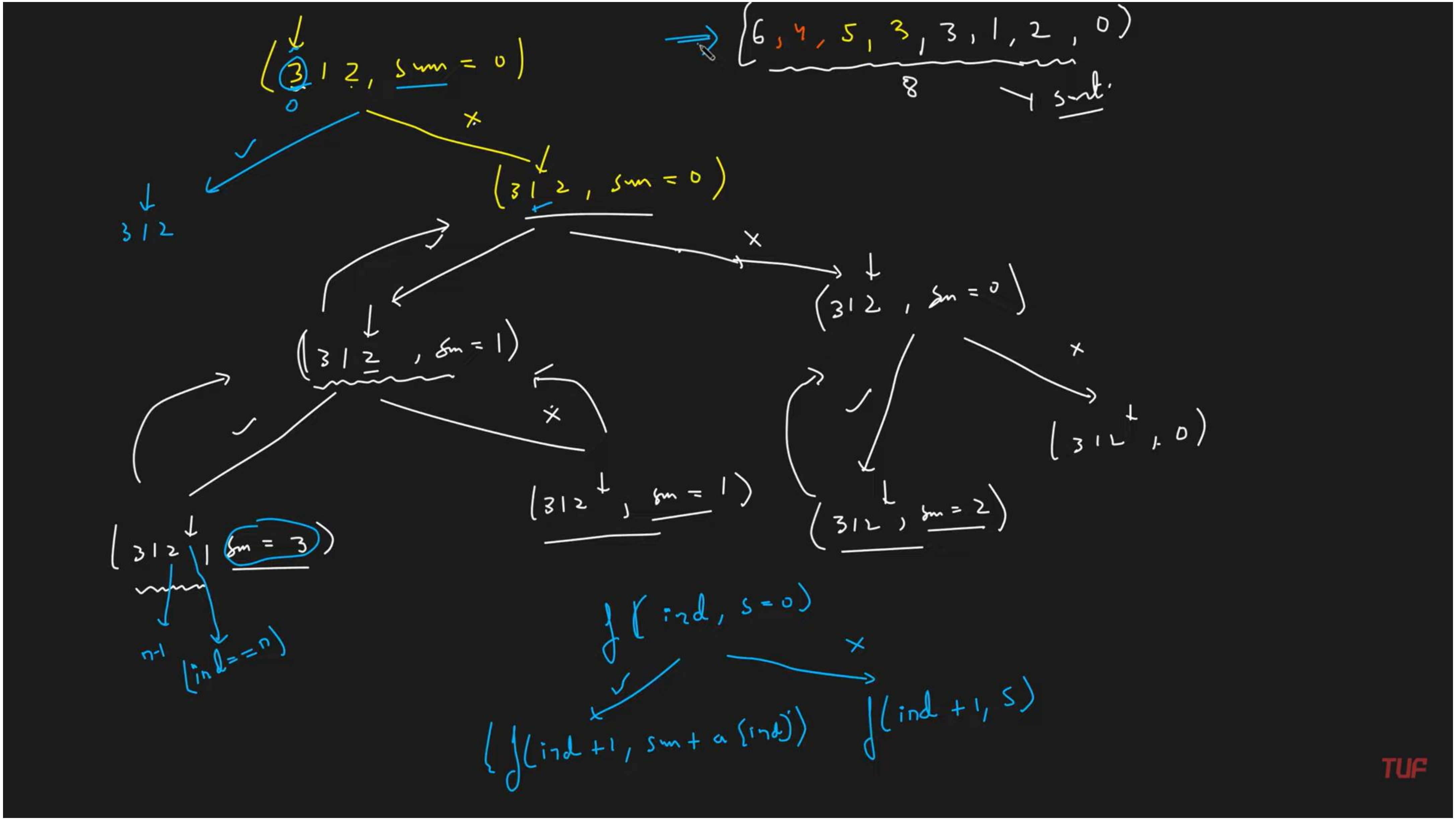

{6, 4, 5, 3, 3, 1
(3 1 2, sum = 0)
×
(3 1 2, sum = 0)
✓
×
(3 1 2, sum = 1)
✓
×
(3 1 2, sum = 3)
(3 1 2, sum = 1)
✓ × ✓
✓ × ×
× ✓ ✓
× ✓ ×

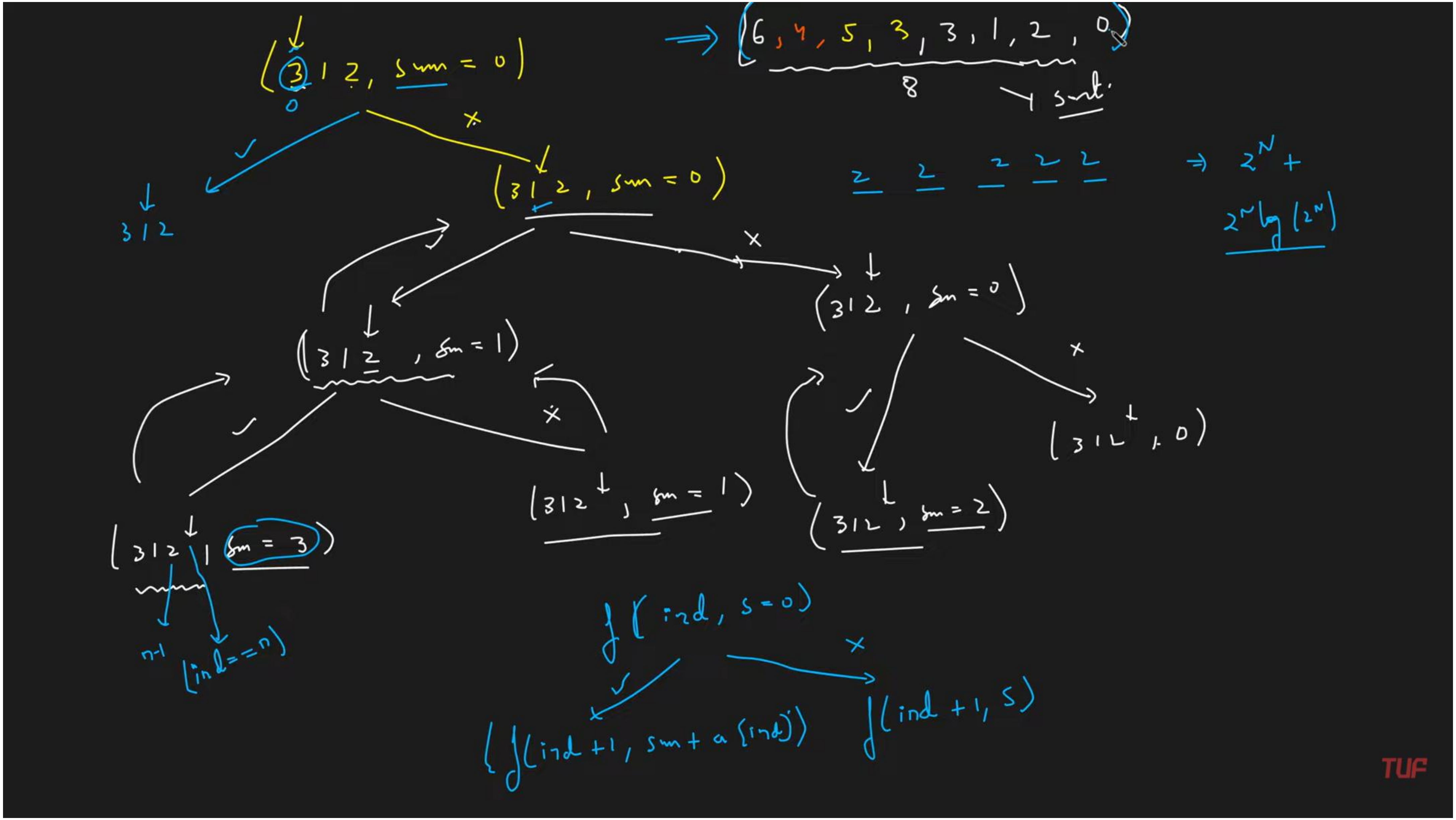

{6, 4, 5, 3, 3, 1
(3 1 2, sum = 0)
(3 1 2, sum = 0)
(3 1 2, sum = 1)
(3 1 2, sum = 3)
(3 1 2, sum = 1)
TUF

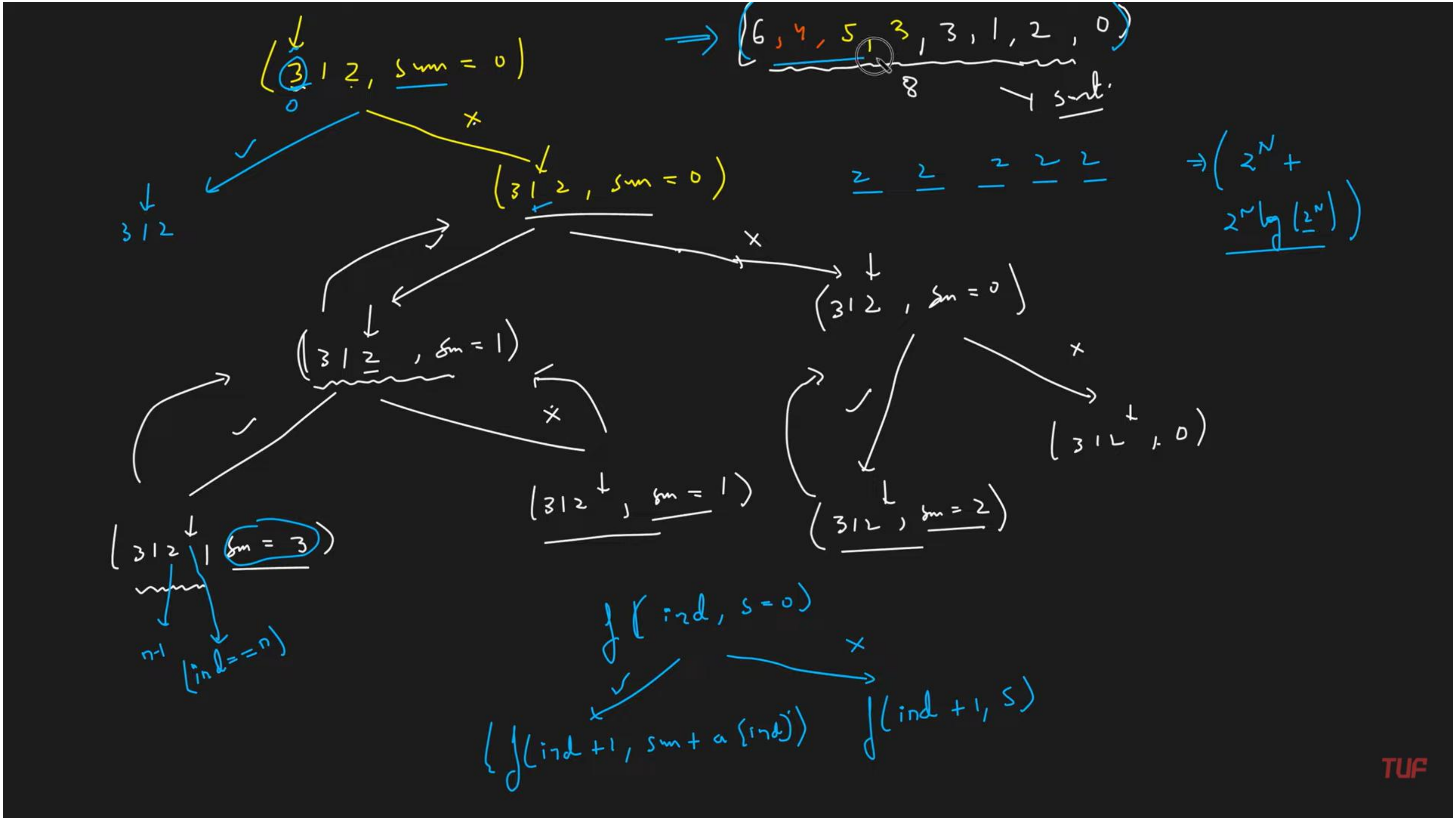

{6, 4, 5, 3, 3, 1, 2
(3 1 2, sum = 0)
×
(3 1 2, sum = 0)
✓
×
(3 1 2, sm = 1)
(3 1 2, sm = 0)
✓
×
✓
×
(3 1 2, sm = 3)
(3 1 2, sm = 1)
(3 1 2, sm = 2)
✓ × ✓
✓ × ×
× ✓ ✓
× ✓ ×
× × ✓

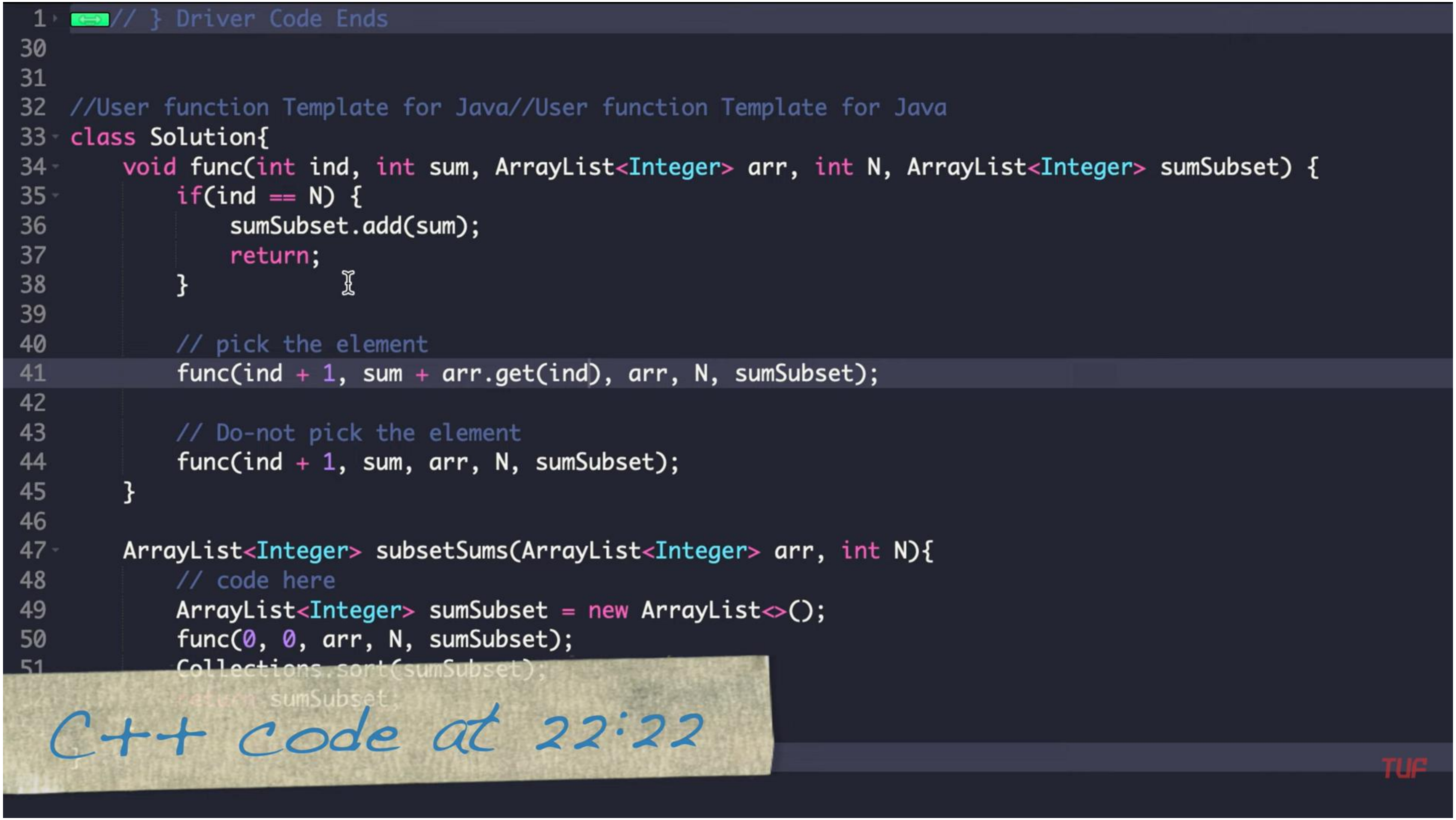

(6, 4, 5, 3, 3, 1, 2, 0)
8
sort
(3 1 2, sum = 0)
×
(3 1 2, sum = 0)
✓
×
(3 1 2, sm = 1)
(3 1 2, sm = 0)
✓
×
✓
×
(3 1 2, sm = 3)
(3 1 2, sm = 1)
(3 1 2, sm = 2)
(3 1 2, 0)


(6, 4, 5, 3, 3, 1, 2, 0)
8
sort
(3 1 2, sum = 0)
0
✓
×
3 1 2
(3 1 2, sum = 0)
(3 1 2, sm = 1)
(3 1 2, sm = 0)
(3 1 2, sm = 3)
(3 1 2, sm = 1)
(3 1 2, sm = 2)
(3 1 2, 0)
f(ind, s=0)
f(ind+1, sum + a[ind])
f(ind+1, s)
TUF


(3 1 2, sum = 0)
0
✓
×
3 1 2
(3 1 2, sum = 0)
×
(6, 4, 5, 3, 3, 1, 2, 0)
8
sort.
(3 1 2, sm = 1)
✓
×
(3 1 2, sm = 0)
✓
×
(3 1 2, sm = 3)
n-1
(ind==n)
(3 1 2, sm = 1)
(3 1 2, sm = 2)
(3 1 2, 0)
f(ind, s=0)
✓
×
f(ind+1, sum + a[ind])
f(ind+1, s)
TUF

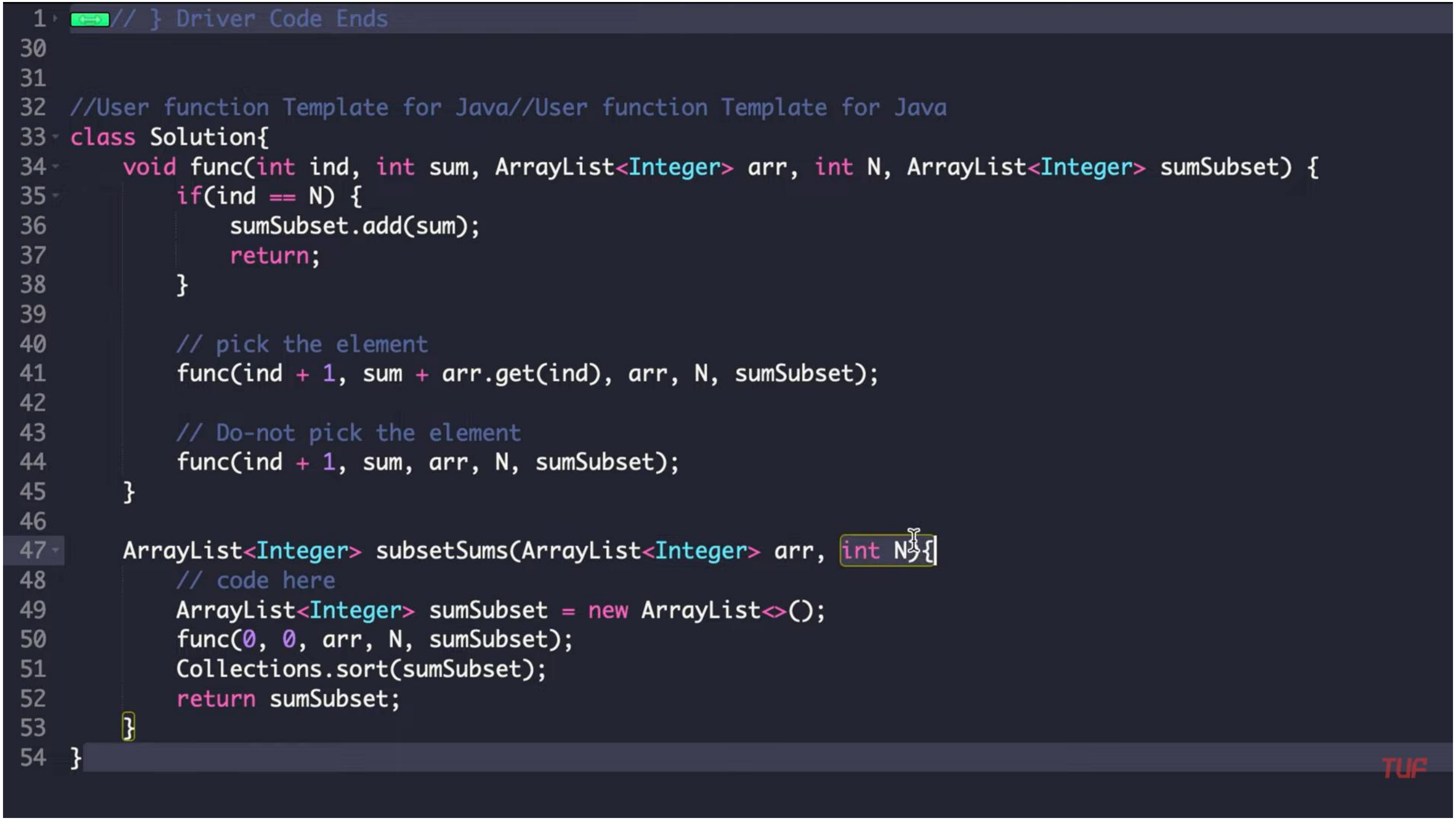

(3 1 2, sum = 0)
0
✓
×
3 1 2
(3 1 2, sum = 0)
(3 1 2, sm = 1)
(3 1 2, sm = 0)
(3 1 2 | sm = 3)
n-1
(ind==n)
(3 1 2, sm = 1)
(3 1 2, sm = 2)
(3 1 2, 0)
⟹ [6, 4, 5, 3, 3, 1, 2, 0]
8
sort.
2 2 2 2 2
⇒ 2^N +
2^N log(2^N)
f(ind, s=0)
{ f(ind+1, sum + a[ind])
f(ind+1, s)
TUF


(3 1 2, sum = 0)
0
✓
×
3 1 2
(3 1 2, sum = 0)
×
(3 1 2, sm = 1)
(3 1 2, sm = 0)
✓
×
✓
×
(3 1 2 | sm = 3)
(3 1 2, sm = 1)
(3 1 2, sm = 2)
(3 1 2, 0)
n-1
(ind == n)
⟹ (6, 4, 5, 3, 3, 1, 2, 0)
8
sort.
2 2 2 2 2
⟹ (2^N + 2^N log (2^N))
f(ind, s = 0)
✓
×
{ f(ind + 1, sum + a[ind])
f(ind + 1, s)
TUF

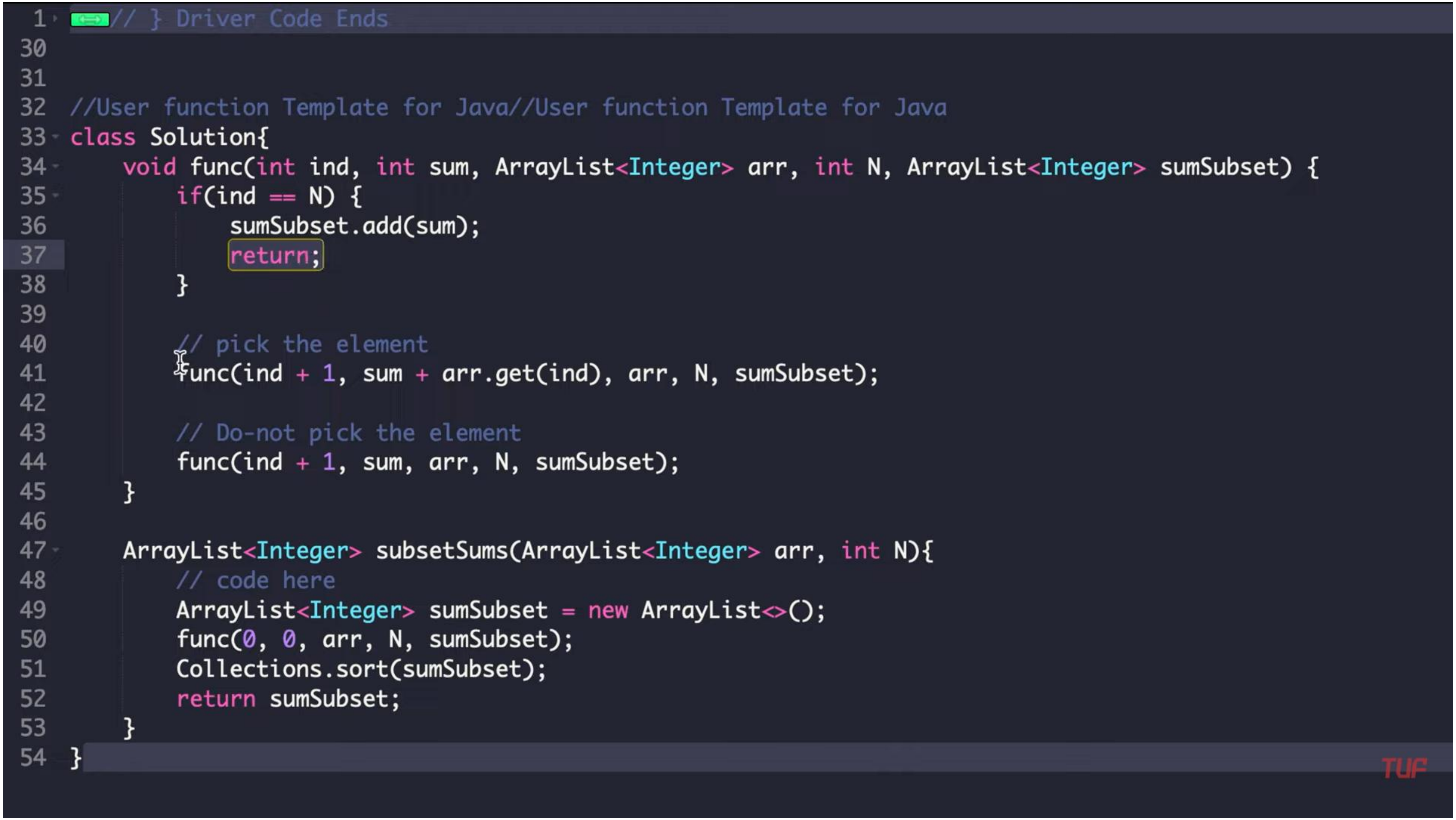


```
// } Driver Code Ends


//User function Template for Java//User function Template for Java
class Solution{
    void func(int ind, int sum, ArrayList<Integer> arr, int N, ArrayList<Integer> sumSubset) {
        if(ind == N) {
            sumSubset.add(sum);
            return;
        }

        // pick the element
        func(ind + 1, sum + arr.get(ind), arr, N, sumSubset);

        // Do-not pick the element
        func(ind + 1, sum, arr, N, sumSubset);
    }

    ArrayList<Integer> subsetSums(ArrayList<Integer> arr, int N){
        // code here
        ArrayList<Integer> sumSubset = new ArrayList<>();
        func(0, 0, arr, N, sumSubset);
        Collections.sort(sumSubset);
        return sumSubset;
```

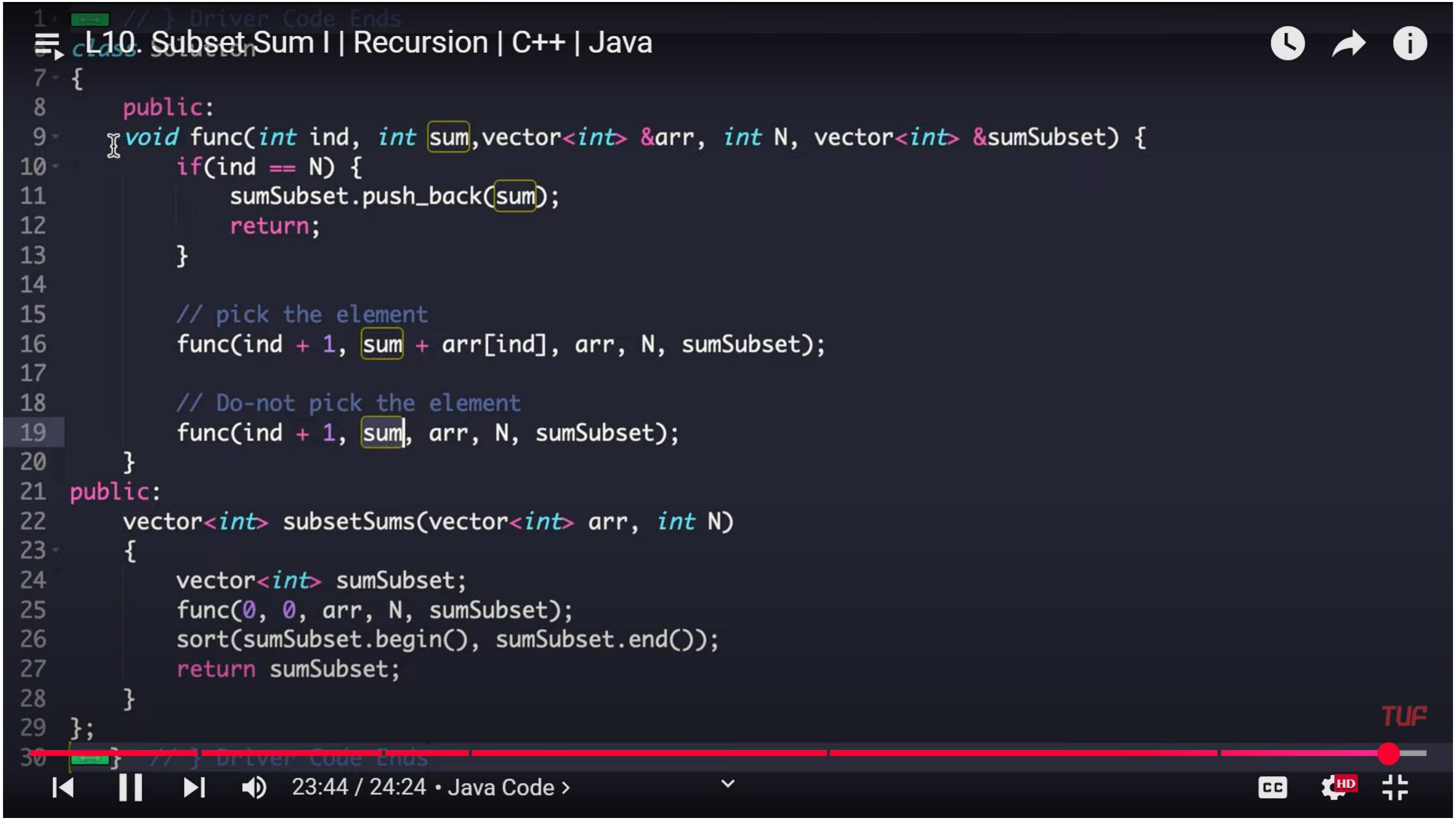
C++ code at 22:22



```java
// } Driver Code Ends


//User function Template for Java//User function Template for Java
class Solution{
    void func(int ind, int sum, ArrayList<Integer> arr, int N, ArrayList<Integer> sumSubset) {
        if(ind == N) {
            sumSubset.add(sum);
            return;
        }

        // pick the element
        func(ind + 1, sum + arr.get(ind), arr, N, sumSubset);

        // Do-not pick the element
        func(ind + 1, sum, arr, N, sumSubset);
    }

    ArrayList<Integer> subsetSums(ArrayList<Integer> arr, int N){
        // code here
        ArrayList<Integer> sumSubset = new ArrayList<>();
        func(0, 0, arr, N, sumSubset);
        Collections.sort(sumSubset);
        return sumSubset;
    }
}
```

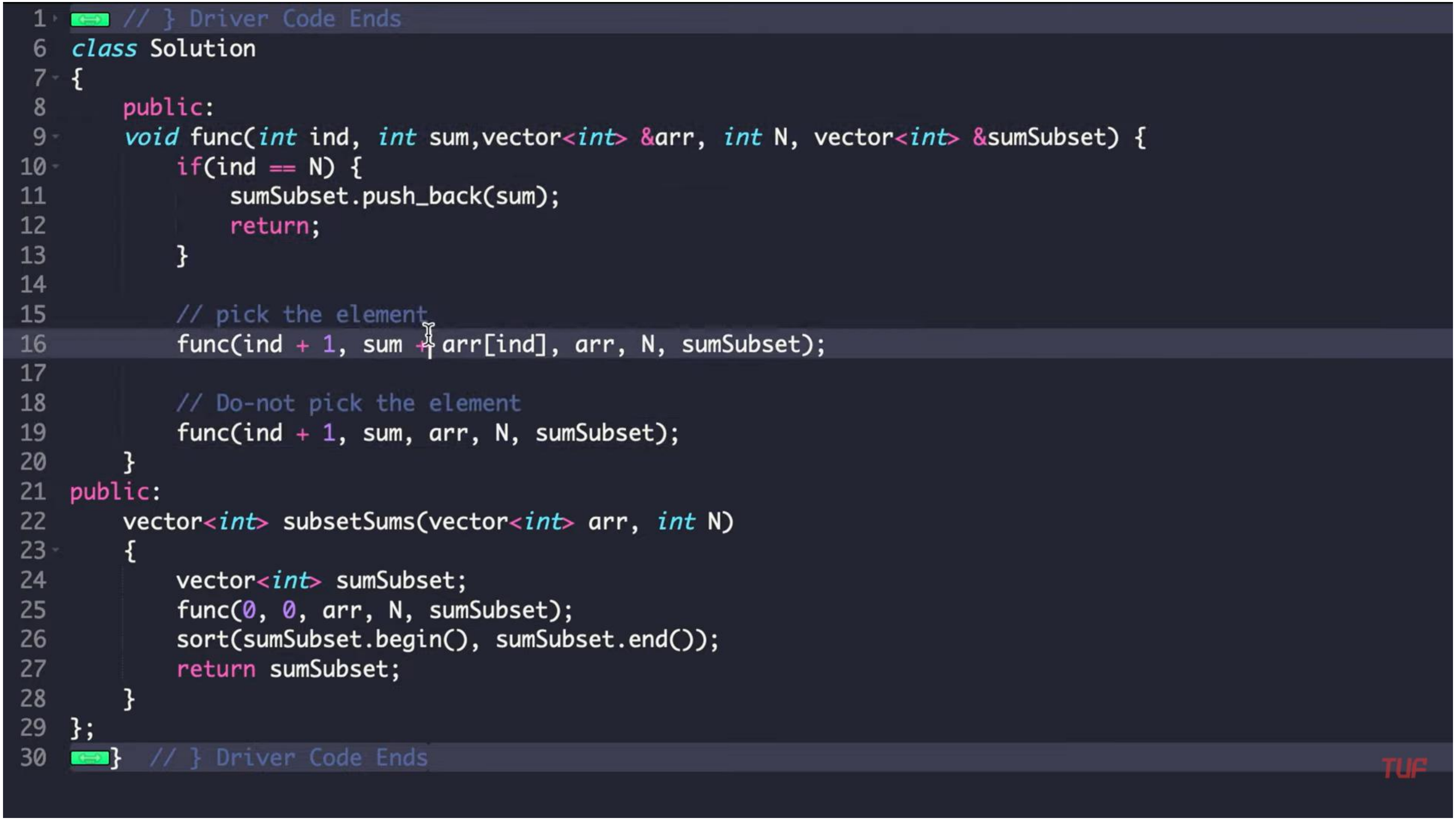


```java
// } Driver Code Ends


//User function Template for Java//User function Template for Java
class Solution{
    void func(int ind, int sum, ArrayList<Integer> arr, int N, ArrayList<Integer> sumSubset) {
        if(ind == N) {
            sumSubset.add(sum);
            return;
        }

        // pick the element
        func(ind + 1, sum + arr.get(ind), arr, N, sumSubset);

        // Do-not pick the element
        func(ind + 1, sum, arr, N, sumSubset);
    }

    ArrayList<Integer> subsetSums(ArrayList<Integer> arr, int N){
        // code here
        ArrayList<Integer> sumSubset = new ArrayList<>();
        func(0, 0, arr, N, sumSubset);
        Collections.sort(sumSubset);
        return sumSubset;
    }
}
```



L10. Subset Sum I | Recursion | C++ | Java
TUF
23:44 / 24:24 • Java Code ›
HD

```
// } Driver Code Ends

class Solution
{
    public:
    void func(int ind, int sum,vector<int> &arr, int N, vector<int> &sumSubset) {
        if(ind == N) {
            sumSubset.push_back(sum);
            return;
        }

        // pick the element
        func(ind + 1, sum + arr[ind], arr, N, sumSubset);

        // Do-not pick the element
        func(ind + 1, sum, arr, N, sumSubset);
    }
public:
    vector<int> subsetSums(vector<int> arr, int N)
    {
        vector<int> sumSubset;
        func(0, 0, arr, N, sumSubset);
        sort(sumSubset.begin(), sumSubset.end());
        return sumSubset;
    }
};
} // } Driver Code Ends
```



```cpp
// } Driver Code Ends
class Solution
{
    public:
    void func(int ind, int sum,vector<int> &arr, int N, vector<int> &sumSubset) {
        if(ind == N) {
            sumSubset.push_back(sum);
            return;
        }

        // pick the element
        func(ind + 1, sum + arr[ind], arr, N, sumSubset);

        // Do-not pick the element
        func(ind + 1, sum, arr, N, sumSubset);
    }
public:
    vector<int> subsetSums(vector<int> arr, int N)
    {
        vector<int> sumSubset;
        func(0, 0, arr, N, sumSubset);
        sort(sumSubset.begin(), sumSubset.end());
        return sumSubset;
    }
};
}  // } Driver Code Ends
```