

Week 4 LIVE 

Backtracking Hard Problems & Doubts

In This Lecture

- ✓ 1. Subsequences of An Array
- 2. Combination Sum Problem
- ✓ 3. Palindromic Partitioning

Subsequences of An Array

Input:

arr[] = [1, 2, 3]

Output:

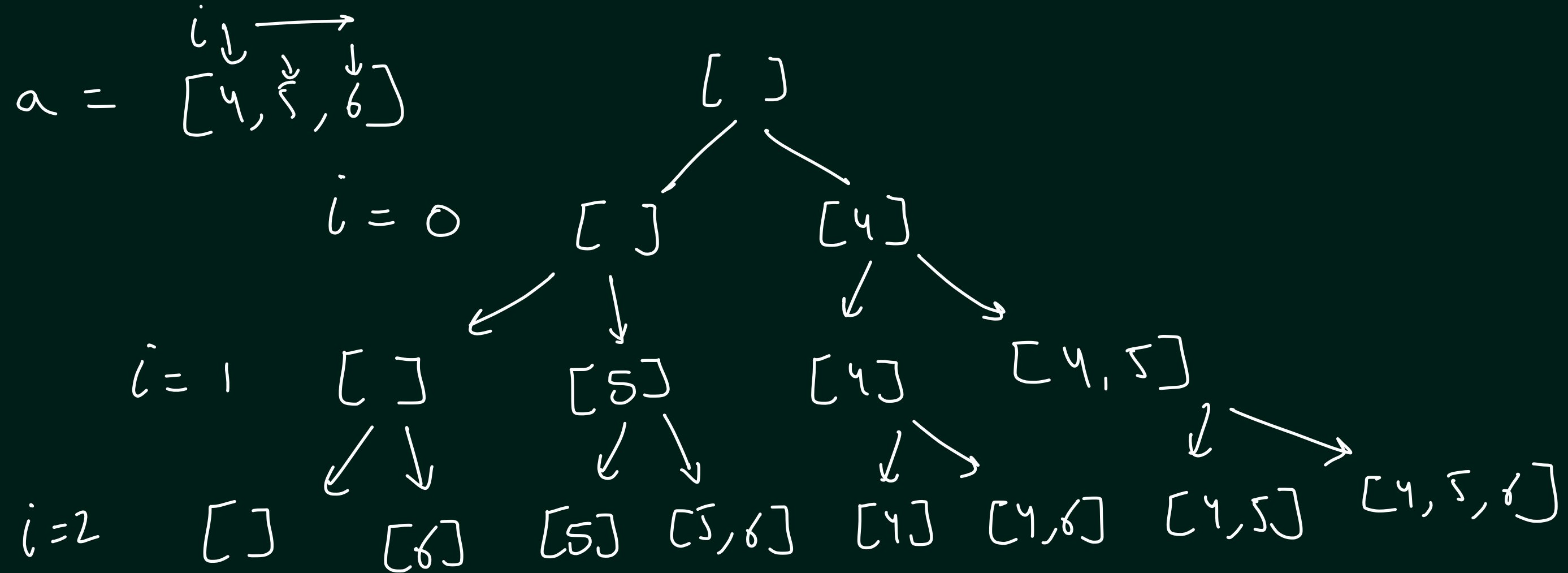
[[], [1], [1, 2], [1, 2, 3], [1, 3], [2], [2, 3], [3]]

$\{a, b, c\} \rightarrow \{a, c\}, \{b, c\}$
 \downarrow
 $\{c\}$

0 or more elements pick
 + including the order

Subarray $\rightarrow \{1, 2, 3, 5\}$
 $\{2, 5\}$ ✓
 $\{5, 2\}$ ✗

Subsequences of An Array



$n = 3$

2^n

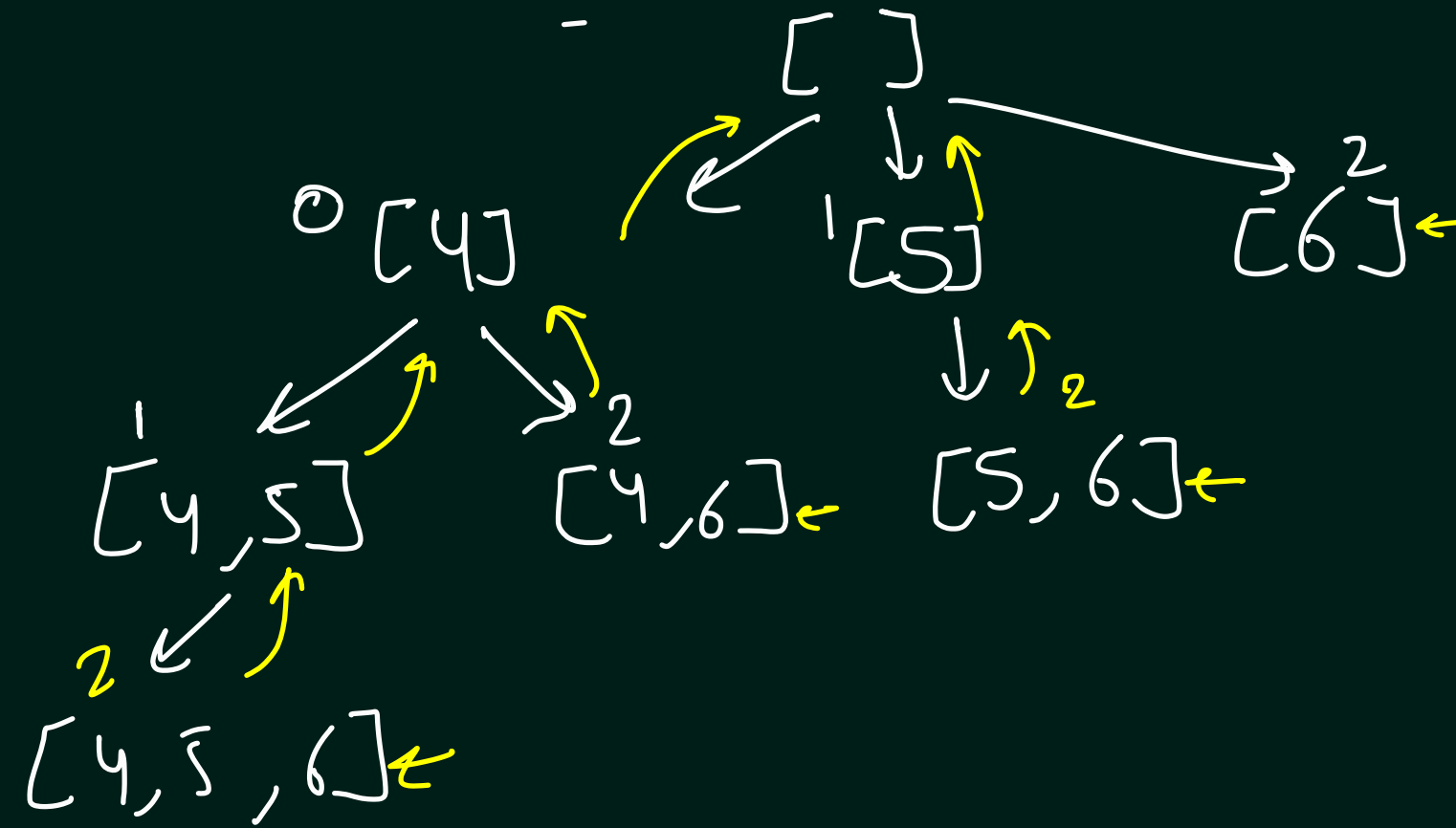
$O(2^n)$

$n \log n \rightarrow 2^n \log_2 2^n \rightarrow O(n \cdot 2^n)$

Subsequences of An Array

→ $a = [4, 5, 6]$

$O(2^n)$



↓ ans

→ []

[4] ←

[4, 5] ←

[4, 5, 6] ←

[4, 6] ←

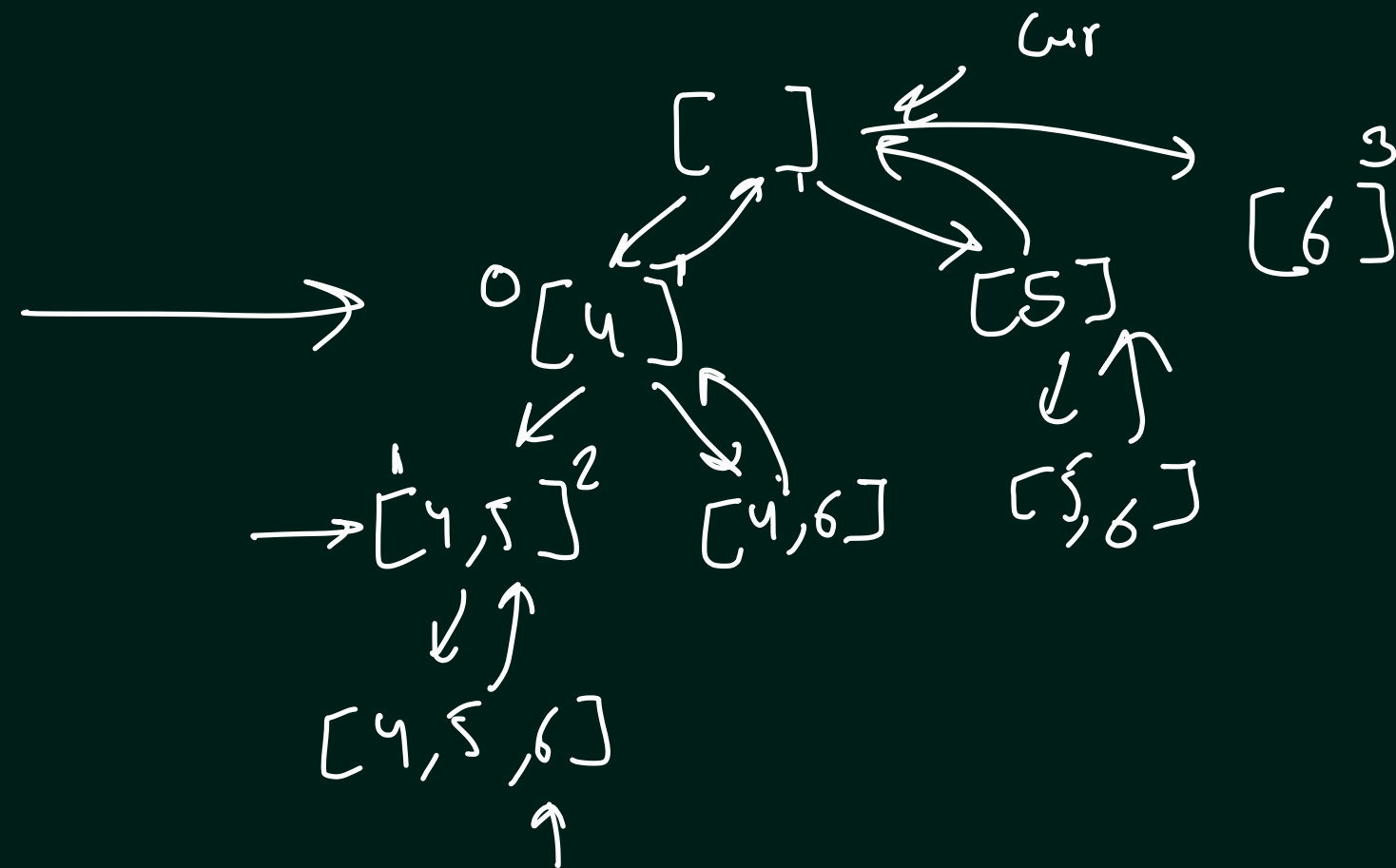
[5]

[5, 6]

[6]

Subsequences of An Array

$a = [4, 5, 6]$



```

static void helperSubsequence(int a[], ArrayList<ArrayList<Integer>> ans,
    int index, ArrayList<Integer> cur) {

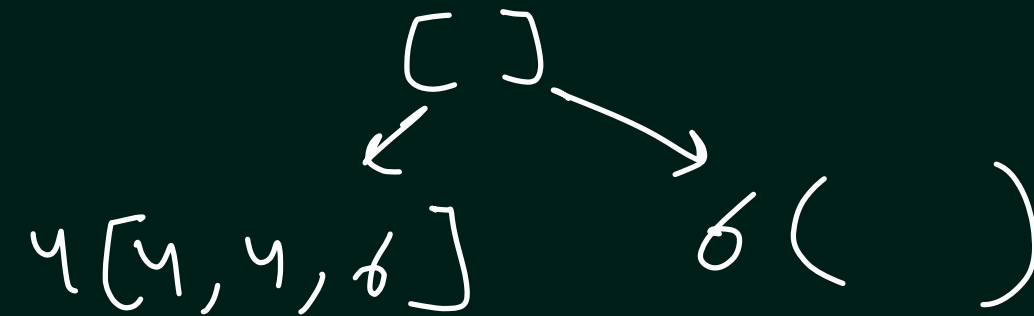
    ArrayList<Integer> curCopy = new ArrayList<>(cur);
    ans.add(curCopy);

    for(int i = index; i < a.length; i++) {
        cur.add(a[i]);
        helperSubsequence(a, ans, i+1, cur);
        cur.remove(index: cur.size()-1); //backtrack
    }
}
    
```

[
 [],
 [4],
 [4, 5]
 [4, 5, 6]
 [4, 6]
 [5]
 [5, 6]
 [6]
]

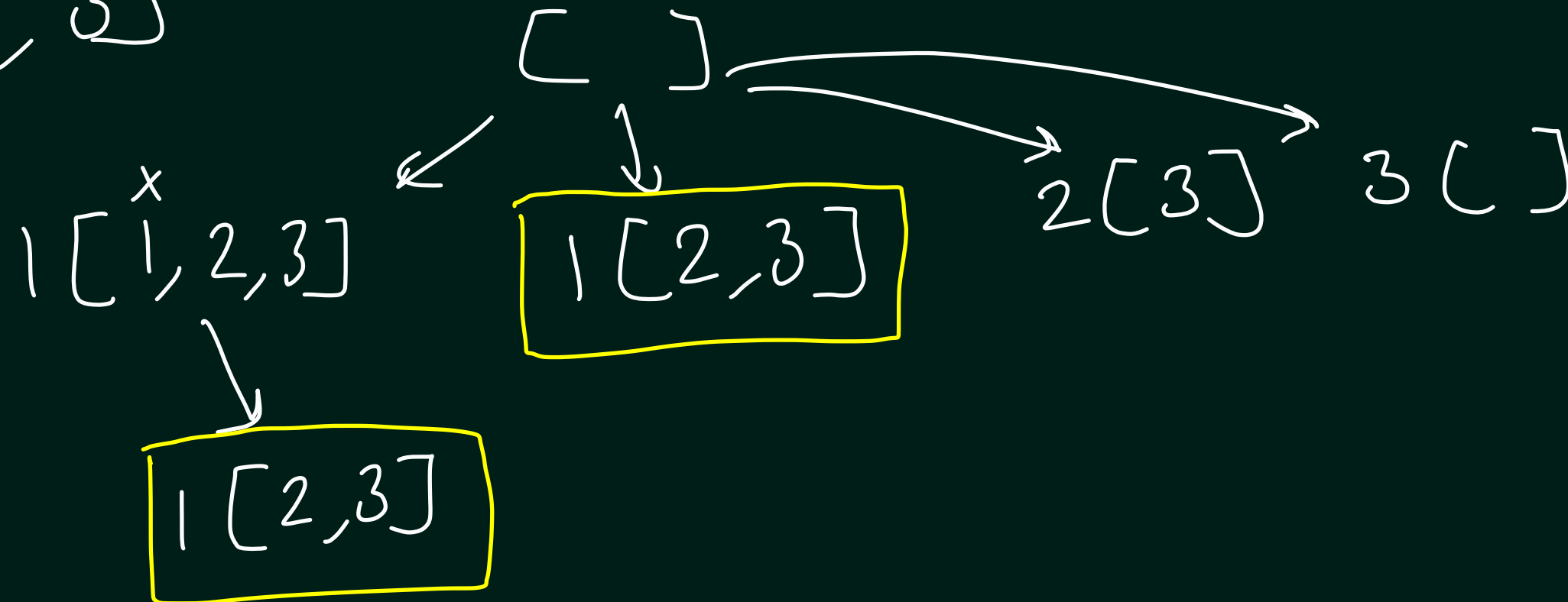
Subsequences of An Array

→ [4, 4^x, 4^x, 6²]



Handling duplicates

↓
[1, 1, 2, 3]



Combination Sum Problem

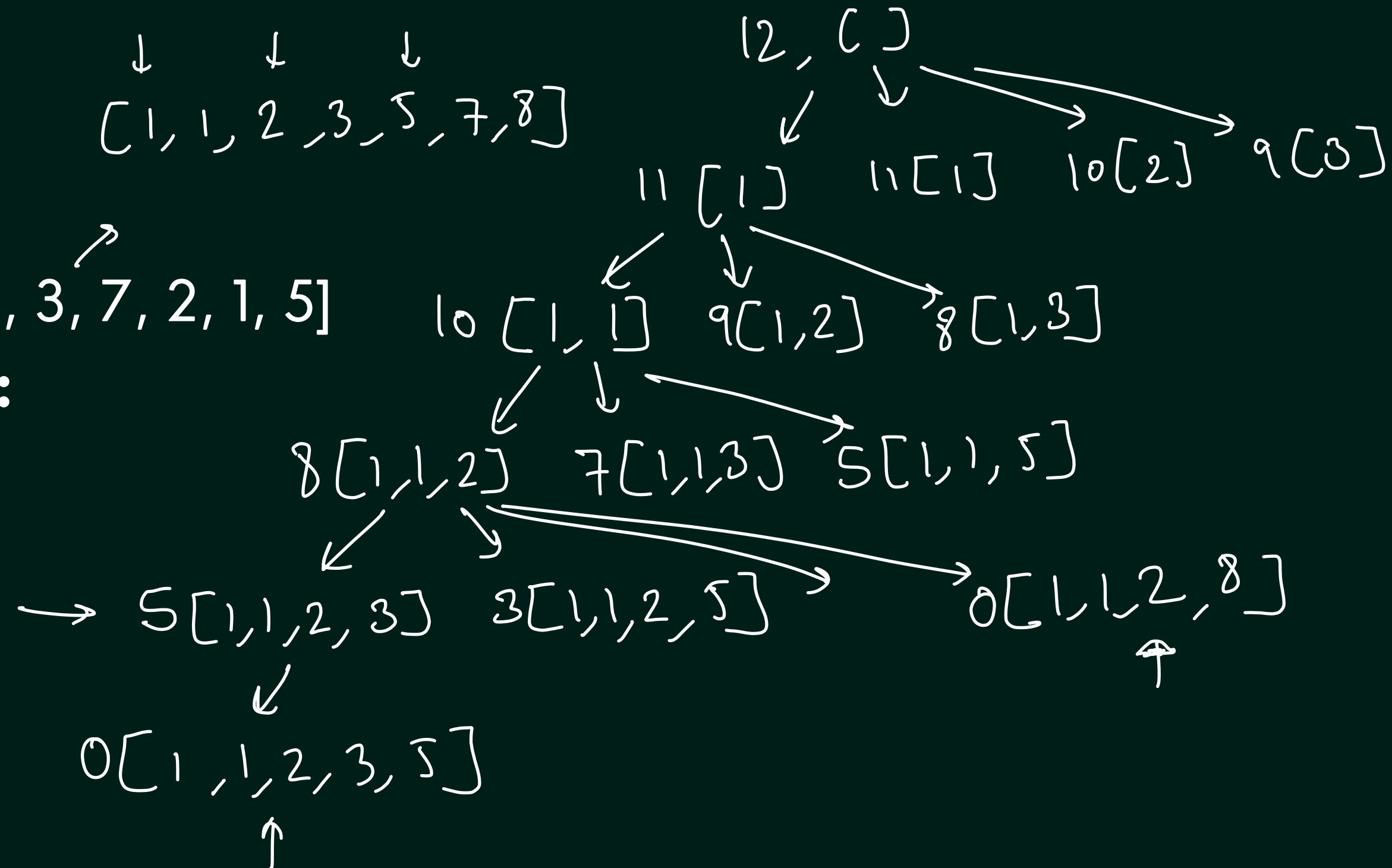
Input:

Target = 12

Candidates = [8, 1, 3, 7, 2, 1, 5]

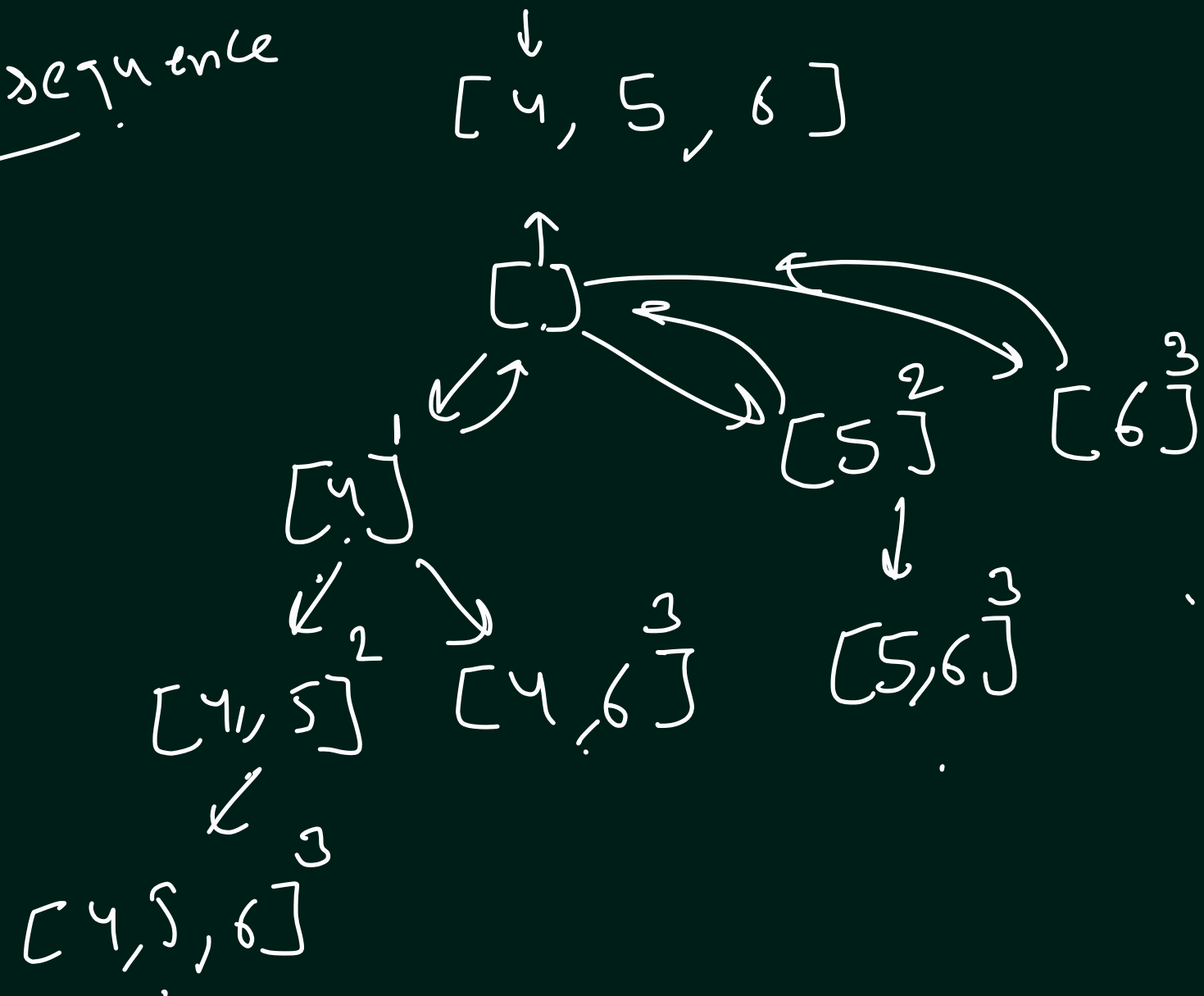
Sample Output:

✓ [[1, 1, 2, 3, 5],
 → [1, 1, 2, 8],
 [1, 1, 3, 7],
 [1, 3, 8],
 [2, 3, 7],
 [5, 7]] →



X Combination Sum Problem

Subsequence



```

static void helperSubsequence(int a[], ArrayList<ArrayList<Integer>> ans,
                             int index, ArrayList<Integer> cur) {

    ArrayList<Integer> curCopy = new ArrayList<>(cur);
    ans.add(curCopy);

    for(int i = index; i < a.length; i++) {

        if(i > index && a[i] == a[i-1]) continue;

        cur.add(a[i]);
        helperSubsequence(a, ans, index: i+1, cur);
        cur.remove(index: cur.size()-1); //backtrack
    }
}
  
```

Palindromic Partitioning

Input:

↳ "aab"

Output:

[["a", "a", "b"],

["a", "a", "b"]]

