RECURSION Concepts & Qns

Video

10

"मैं, DSA की शपथ लेता हूँ कि मैं जो पढ़ाउगा बहोत अच्छे से पढ़ाउगा।"

Facebook
Instagram → codestorywithMIK

(Twitter) → CSwithMIK

codestorywithMIK → ☎

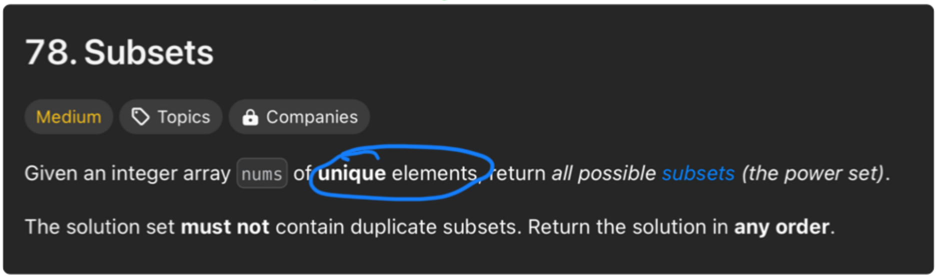
Motivation
(भाषण)

" Don't be discouraged by the complexity of data & algorithms. Break down the concepts into smaller, manageable pieces. "

"That's how you can crack any tough Problem...

#CodestorywithMIK ...

MICROSOFT



78. Subsets

Medium | Topics | Companies

Given an integer array `nums` of **unique** elements, return *all possible* subsets *(the power set)*.

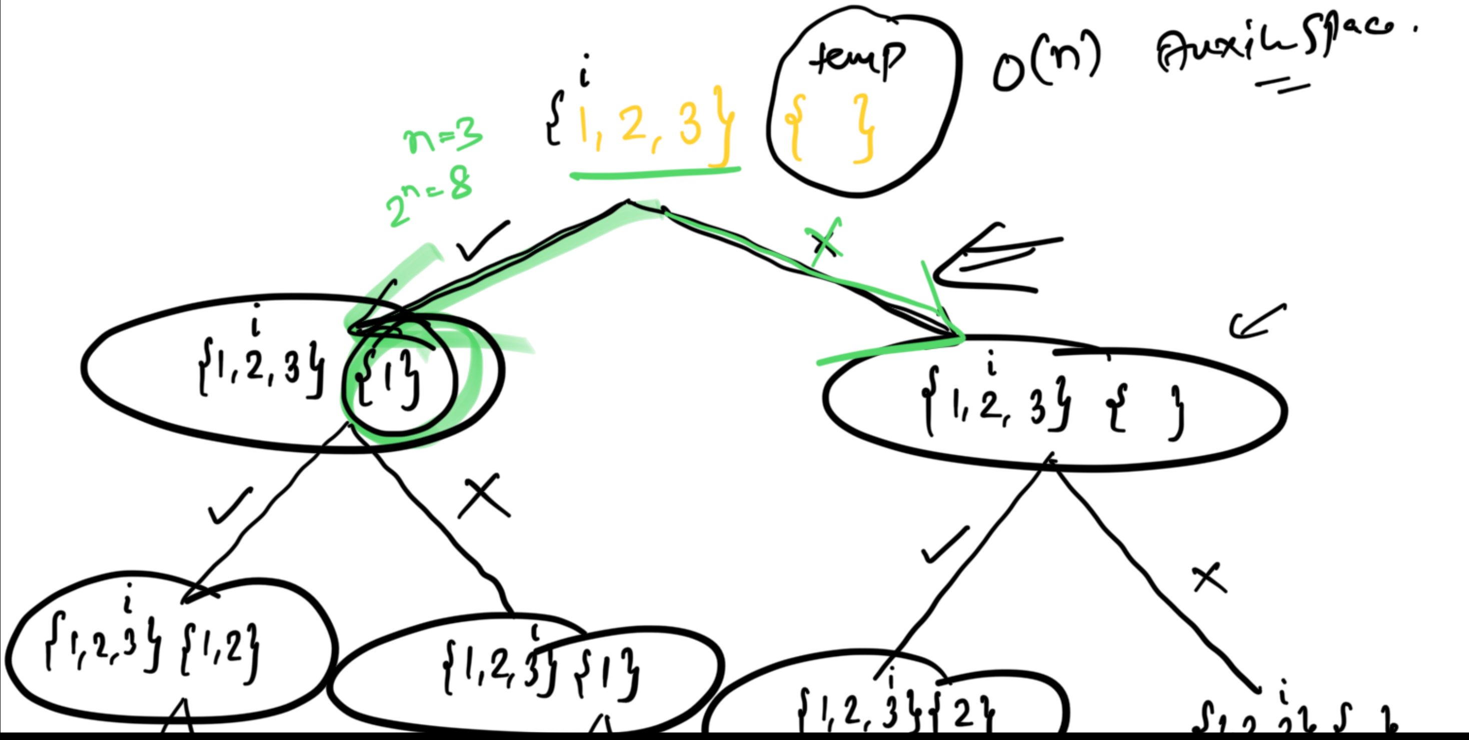
The solution set **must not** contain duplicate subsets. Return the solution in **any order**.

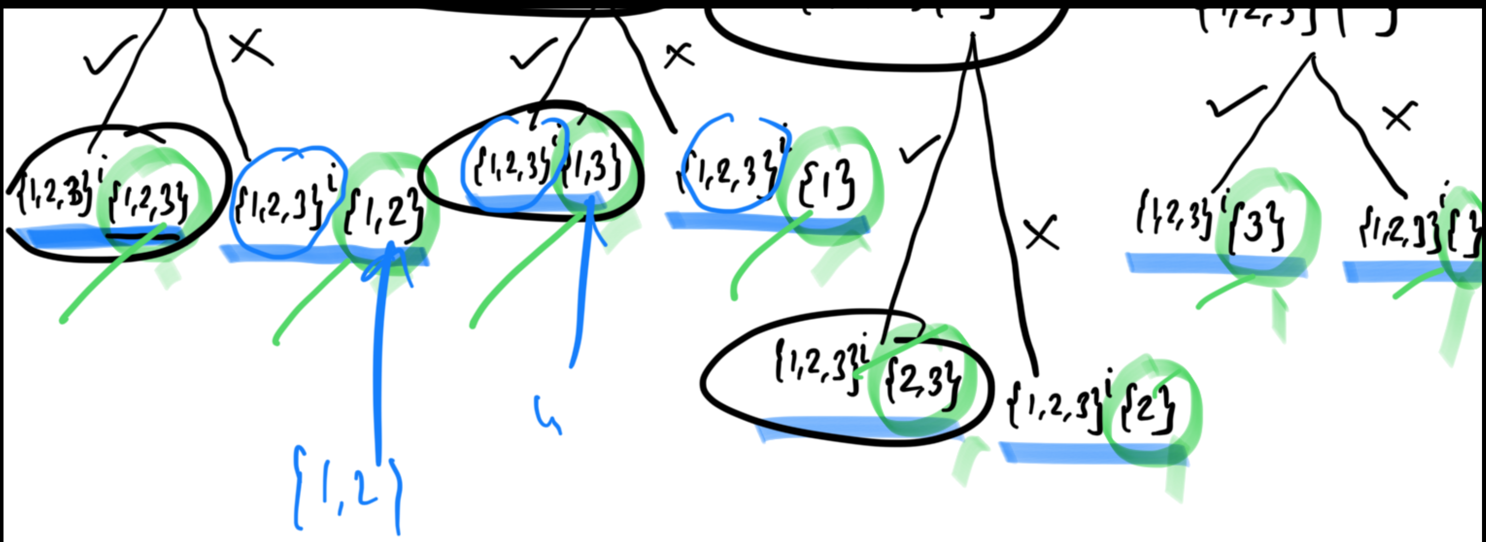
Example :- nums = {1, 2, 3}

Output = [{ }, {1}, {2}, {3}, {1,2}, {1,3}, {2,3}, {1,2,3}]

take
not-take

Options → Recursion
Hint.





(•) Tree Diagram

↳ options

↳ Base Case

↳ System stack space → O(n)

↳ Auxi → O(n) to store subsets

↳ T.C = (2^n)

↓

-> Every element has 2 options.

-> we've n elements.

$2^n * n$

Space.

Recursion Magic.

(Trust).

```
Solve (nums , i , temp)    {
    if (i >= nums.size()) {
        result.push_back (temp);
    }
    temp.push_back (nums [i]);    //Take
    Solve (nums, i+1 , temp);     // Trust
    temp.pop_back ();             //not-take
    Solve (nums, i+1, temp);      //Trust
}
```

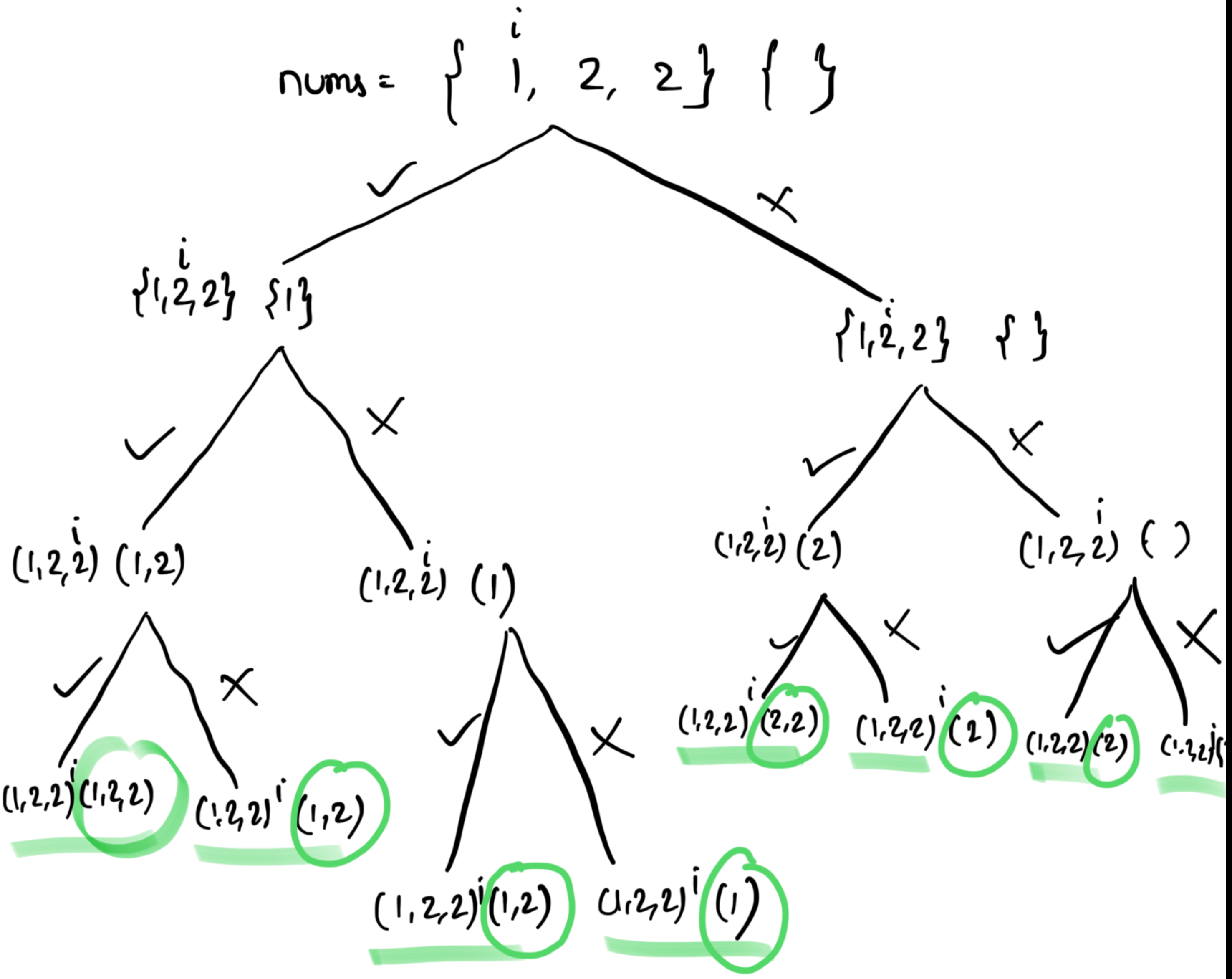
Backtrack

leap of faith.

what if we have

duplicate elements :-

duplicate elements



(1, 2, 2)

(1,2) * duplicate

(1)

(2,2)

(2) * dupli-