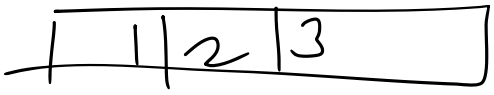
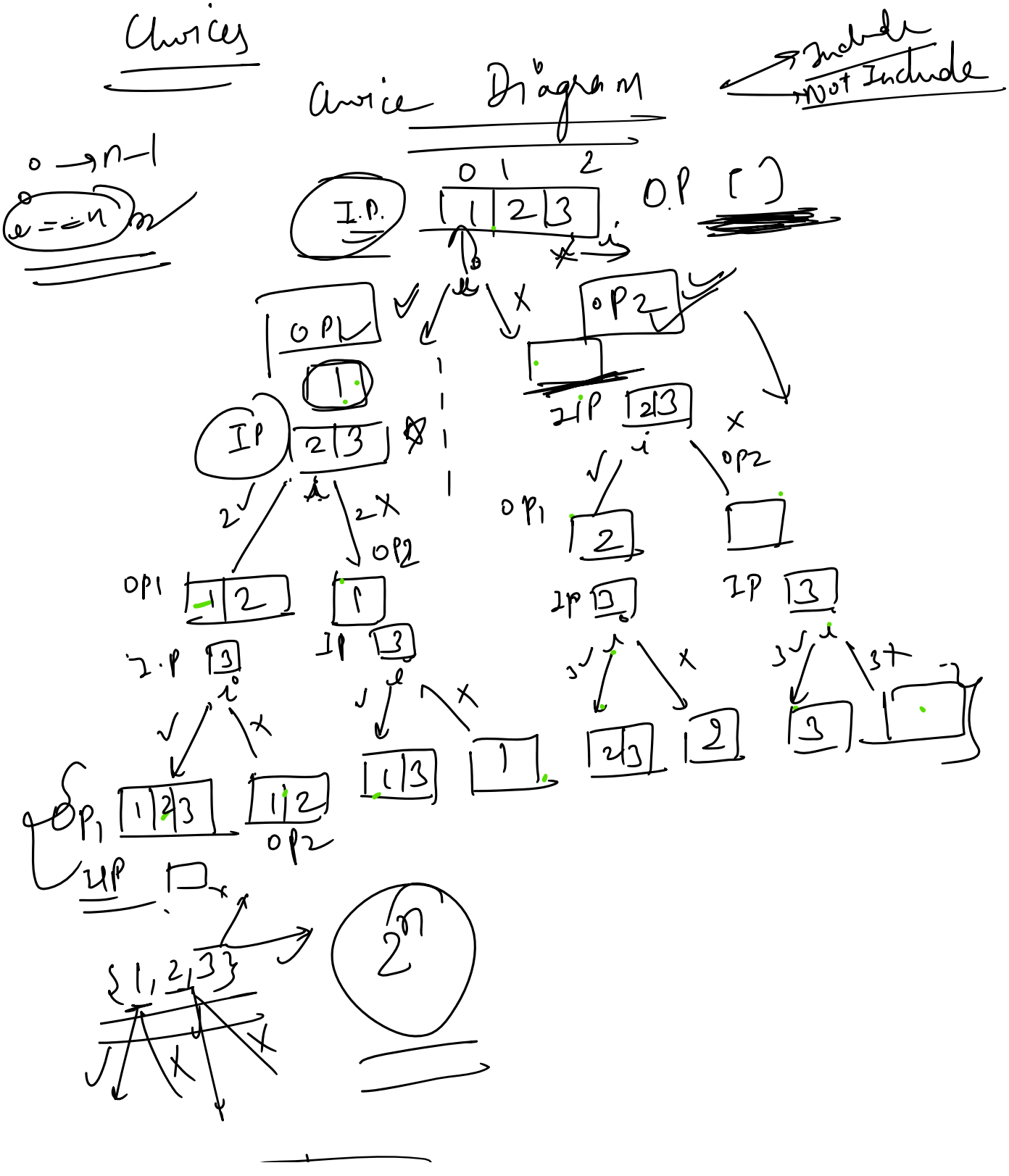
Day 2

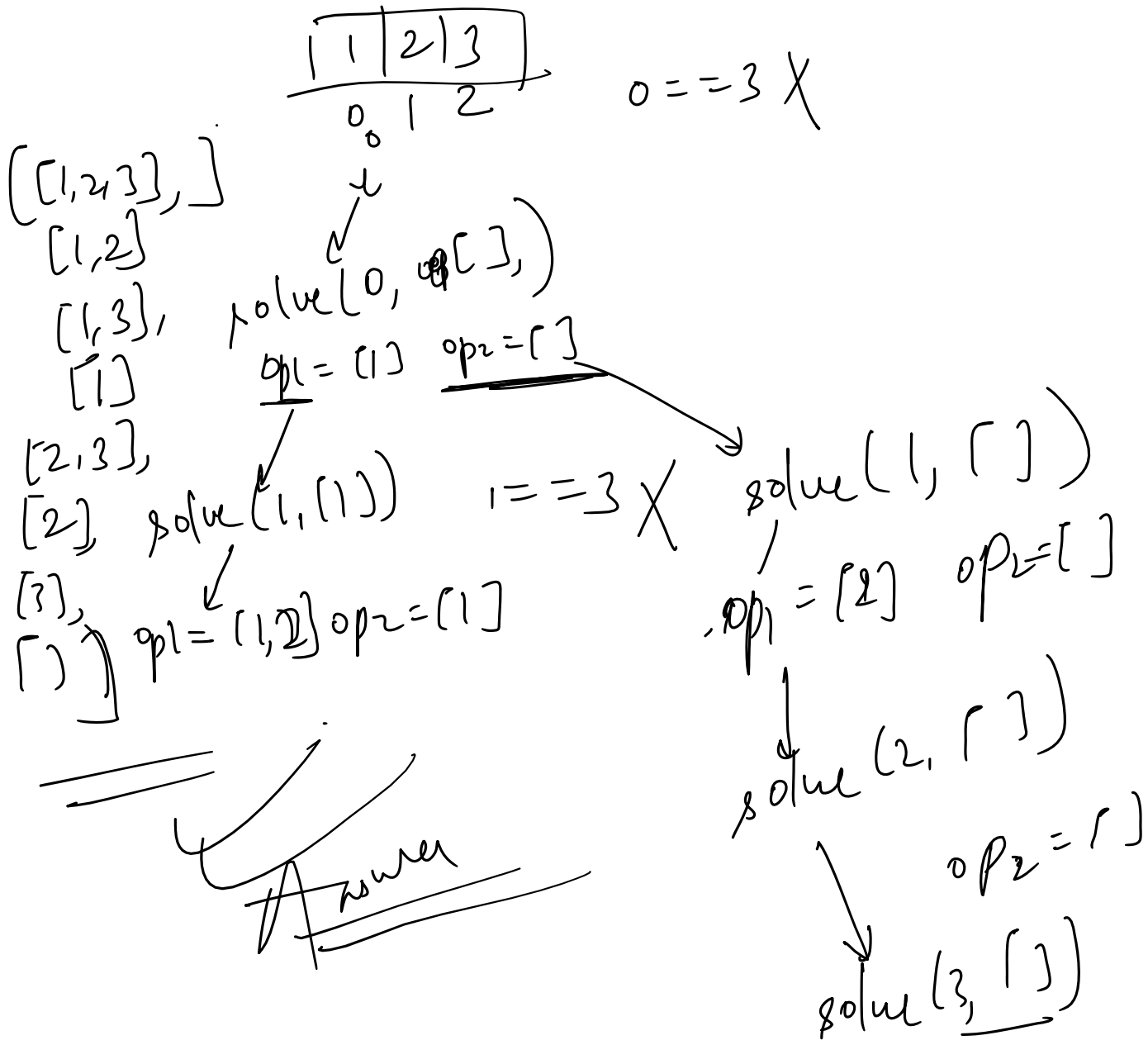
22 November 2022 19:16

Subsets



Choices



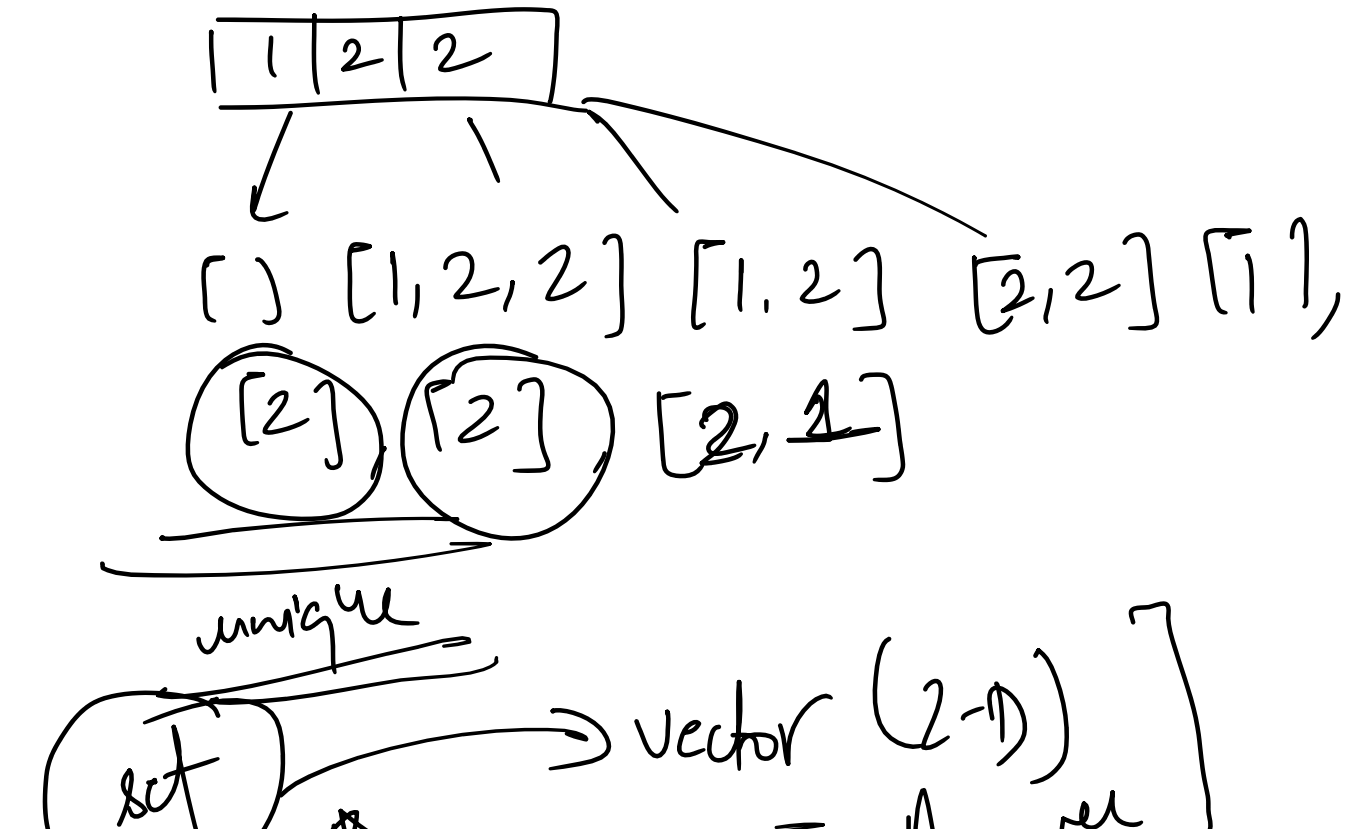

1 1 2 3
0 1 2
0 == 3 X
[[1,2,3],]
[1,2]
[1,3],
[1]
[2,3],
[2],
[3],
[]
solve(0, [])
op1 = [1] op2 = []
solve(1, [1])
1 == 3 X
op1 = [1,2] op2 = [1]
solve(1, [])
op1 = [2] op2 = []
solve(2, [])
op2 = [1]
solve(3, [])
Answer

```cpp
void solve(vector<int> &nums,vector<int> op,vector<vector<int>> &ans, int i){
    if(i == nums.size()){
        ans.push_back(op);
        return;
    }
    // choice diagram
    vector<int> op1 = op;
    vector<int> op2 = op;
    // include
    op1.push_back(nums[i]);
    solve(nums,op1,ans,i+1);
    // not included
    solve(nums,op2,ans,i+1);
}
vector<vector<int>> subsets(vector<int>& nums) {
    //anwser 2-d vector
    vector<vector<int>> ans;
    int i = 0;
    vector<int> op;
    solve(nums,op,ans,i);
    return ans;
}
```

T.C- O(2^N)
S.C- O(N)

N=3

Subset 2



set → vector (( )) Answer

Subset1

T.C – (2^N + log X)

N → set X

$X = 2^N$

focus

| 1 | 2 | 2 |

① sort → equal elements together

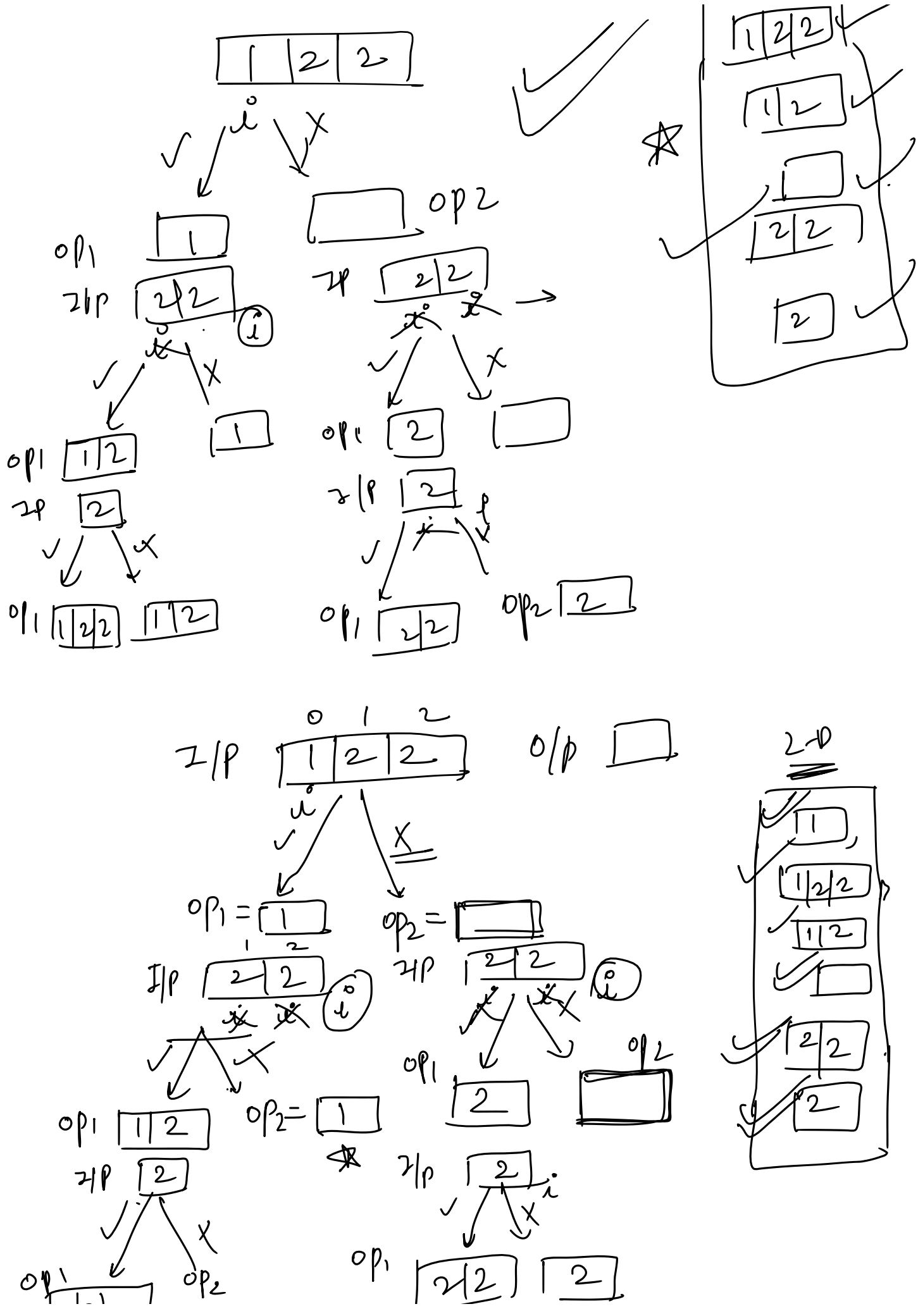
// Include → ✓

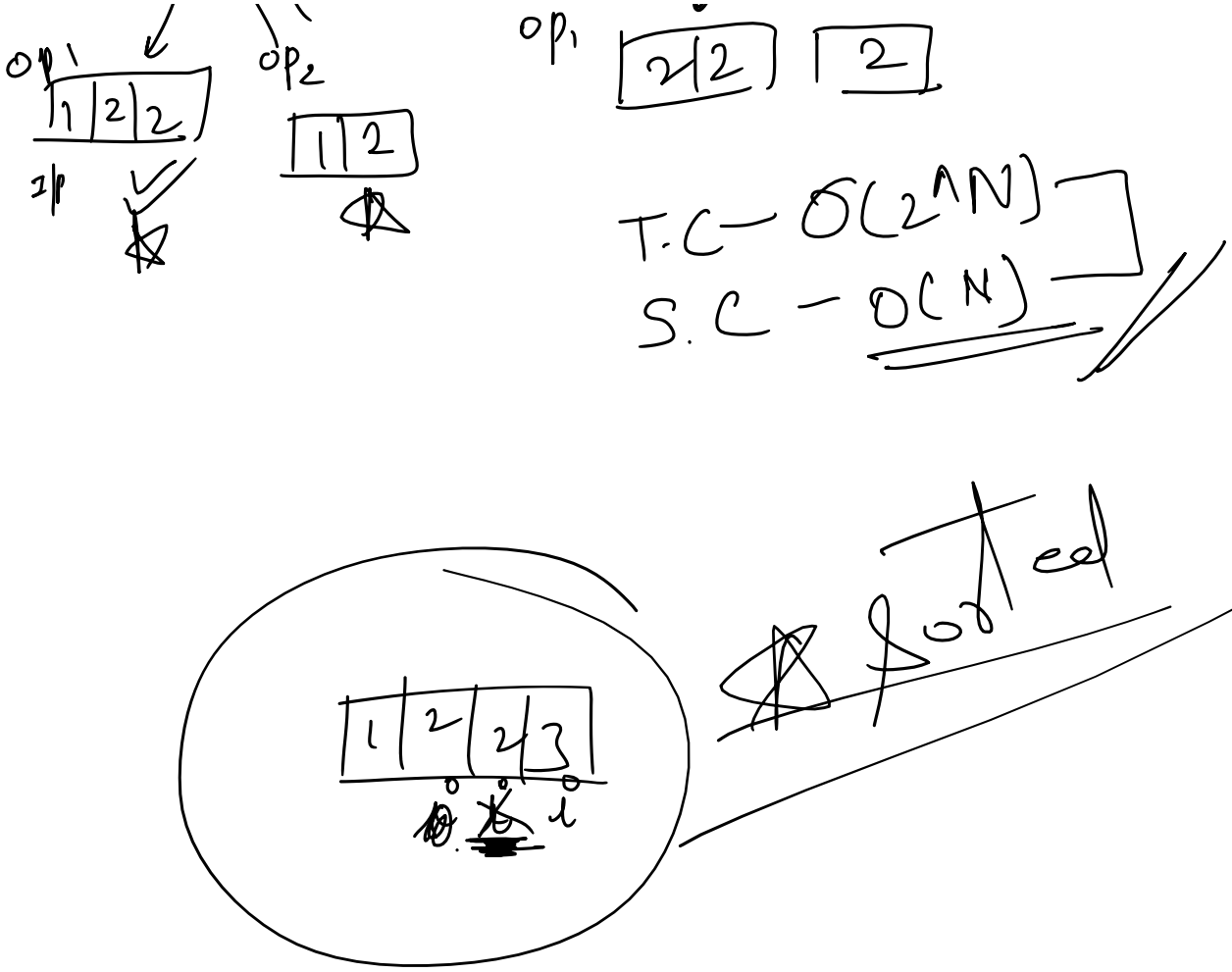
// Not Include // × Ignore

[1], ✓

[1 | 2 | 2], ✓

[1 | 2 | 2]






```cpp
void solve(vector<int> &nums,vector<int> op,vector<vector<int>> &ans, int i){
    if(i == nums.size()){
        ans.push_back(op);
        return;
    }
    // choice diagram
    vector<int> op1 = op;
    vector<int> op2 = op;
    // include
    op1.push_back(nums[i]);
    solve(nums,op1,ans,i+1);
    // not included
    while(i+1 < nums.size() && nums[i] == nums[i+1]){
        i++;
    }
    solve(nums,op2,ans,i+1);
}
vector<vector<int>> subsetsWithDup(vector<int>& nums) {
    sort(nums.begin(),nums.end());
    vector<vector<int>> ans;
    int i = 0;
    vector<int> op;
    solve(nums,op,ans,i);
    return ans;
}
```