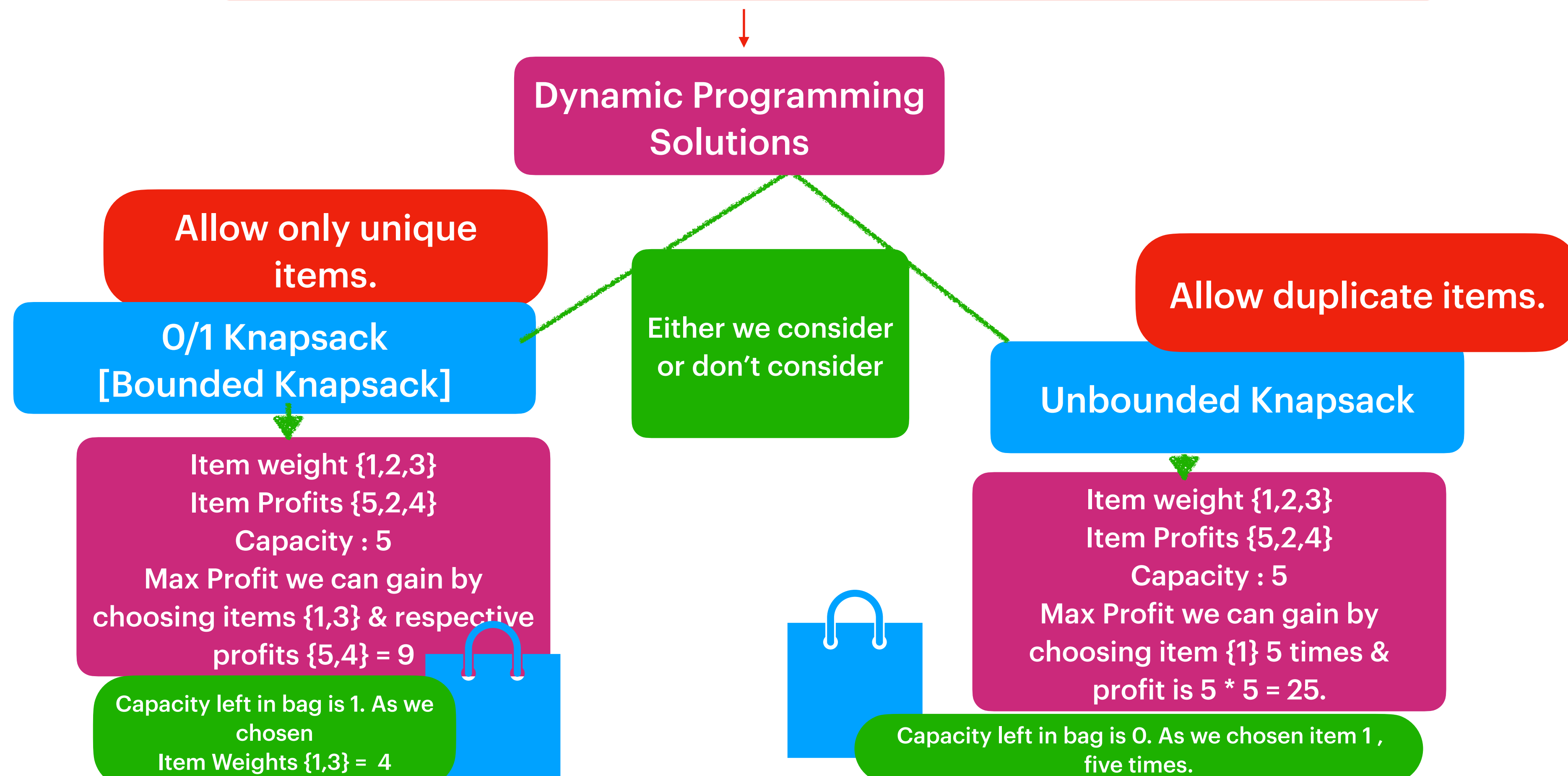


Example

Problem Statement : Every Knapsack has the capacity, we should choose the items , which can give higher profit, by considering capacity in mind.

example : itemsWeight = {1,2,3} itemsProfits = {5,2,4} capacity : 5



416. Partition Equal Subset Sum

Medium

👍 7299

💬 115

❤ Add to List

📄 Share

Given a **non-empty** array `nums` containing **only positive integers**, find if the array can be partitioned into two subsets such that the sum of elements in both subsets is equal.

Example 1:

Input: `nums = [1,5,11,5]`

Output: `true`

Explanation: The array can be partitioned as `[1, 5, 5]` and `[11]`.

Example 2:

Input: `nums = [1,2,3,5]`

Output: `false`

Explanation: The array cannot be partitioned into equal sum subsets.

Constraints:

- `1 <= nums.length <= 200`
- `1 <= nums[i] <= 100`

There are n elements in an array, we partitioning into two subSets. Then each element (:index) can be included either of one subSet but not in both the subSets:

So In a Equal Subset Sum partition.

If we find out 1st sub array sum, which is equals $\text{totalSum}/2$.

2nd sub array sum will obviously equals to $\text{totalSum}/2$.

`nums : [1,5,11,5] :: sum:22`

`{SubArray1: 1,5,5} --> {SubArray2:11}`
`sum/2 ::11 --> sum/2::11`

`[1,5,11,5]` can this be partitioned to two equal subset sum:

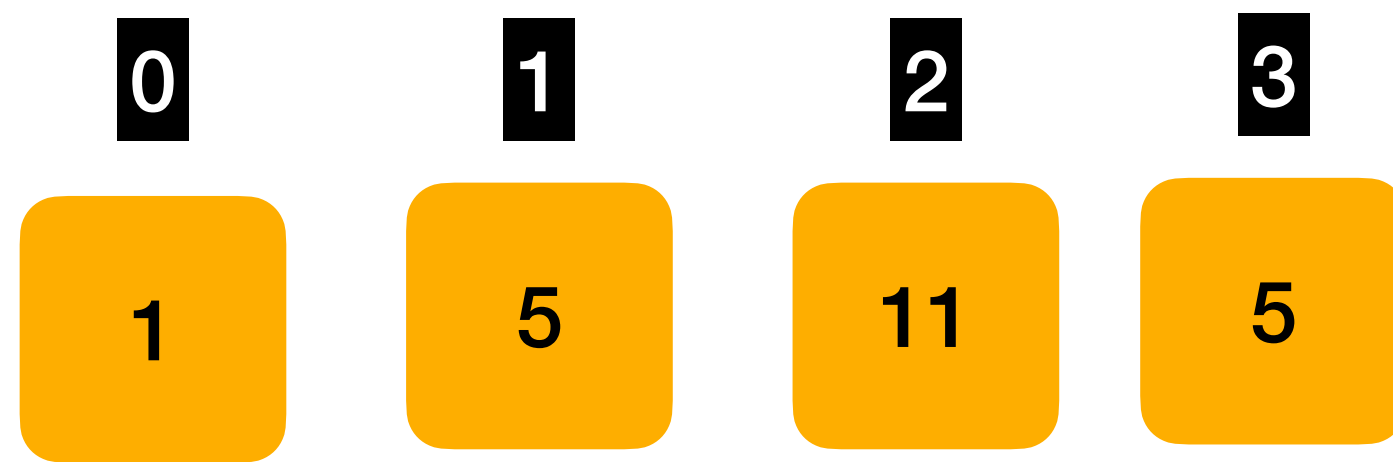
`[1,5,5]sum:11 == [11]sum:11`
`11 = 11 True`

`[1,2,3,5] --> (X)` Summation of input array is 11 :
If the sum is odd we can not make equal partitions.

`[3,3,5,3] --> (X) False Sum : 14`

`[3,3] sum:6 --> [5,3] sum:8`

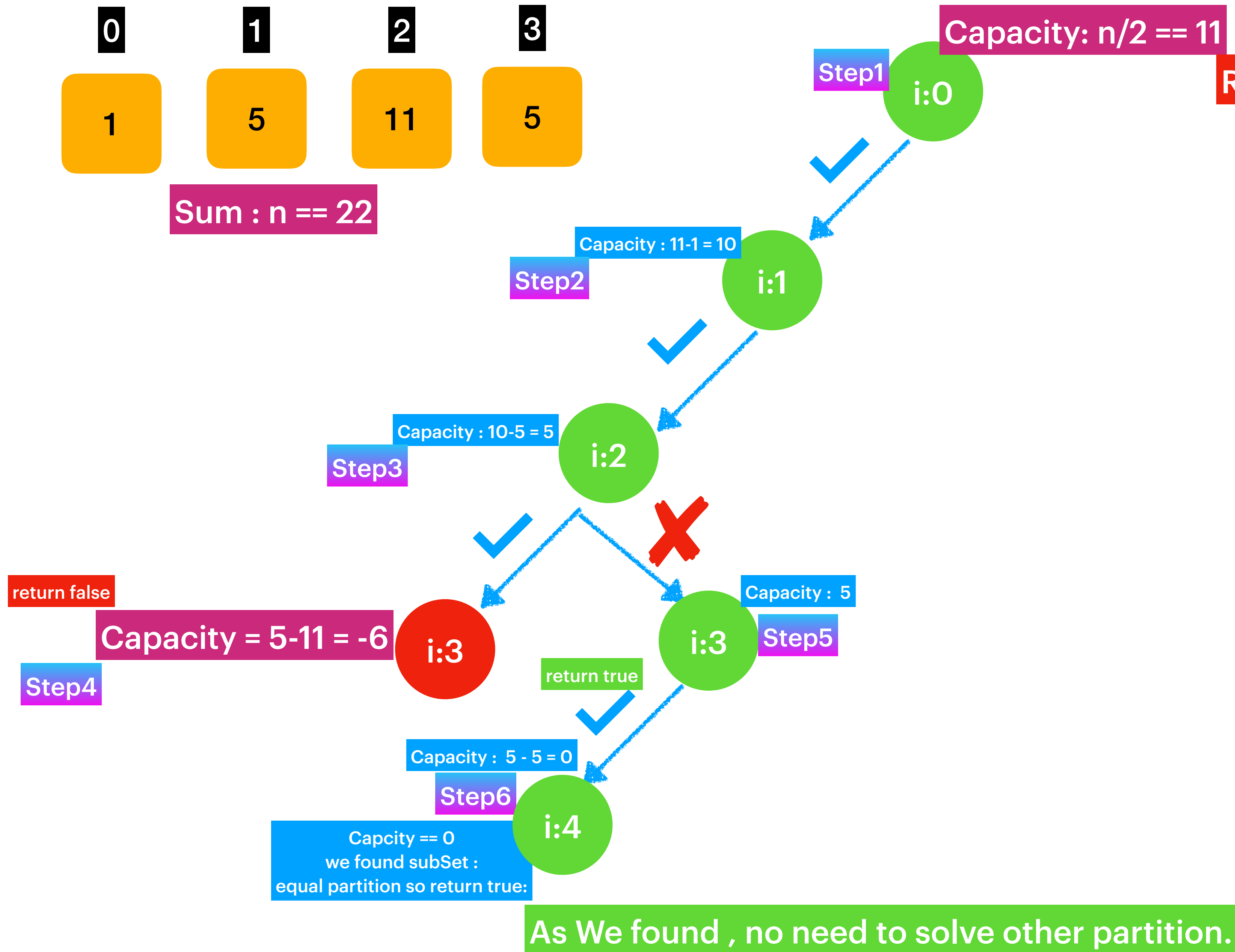
`[3,3,3]sum:8 --> [5]sum:5`

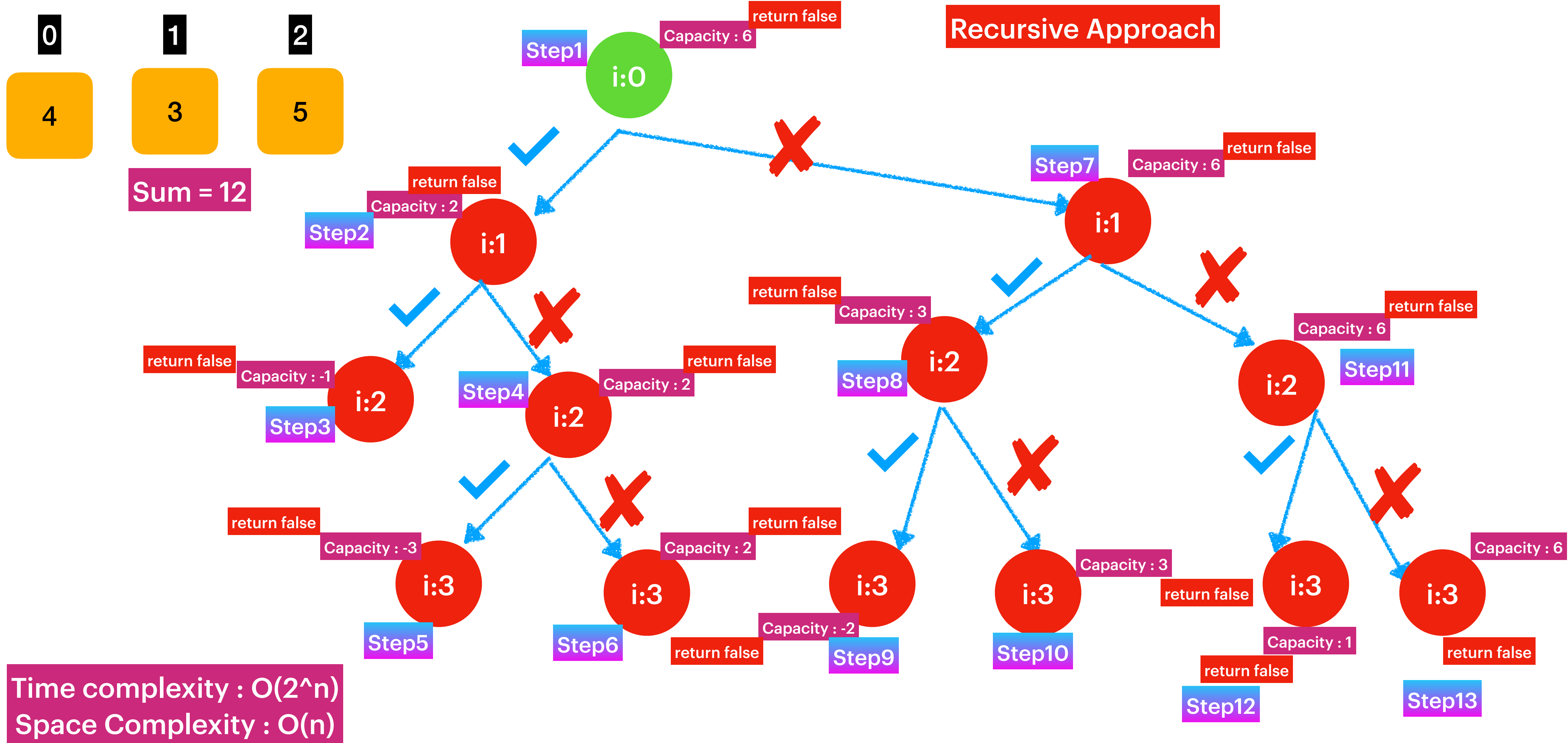


Sum : n == 22

Capacity: $n/2 == 11$

Recursive Approach





Non Of the Recursive call returns true: