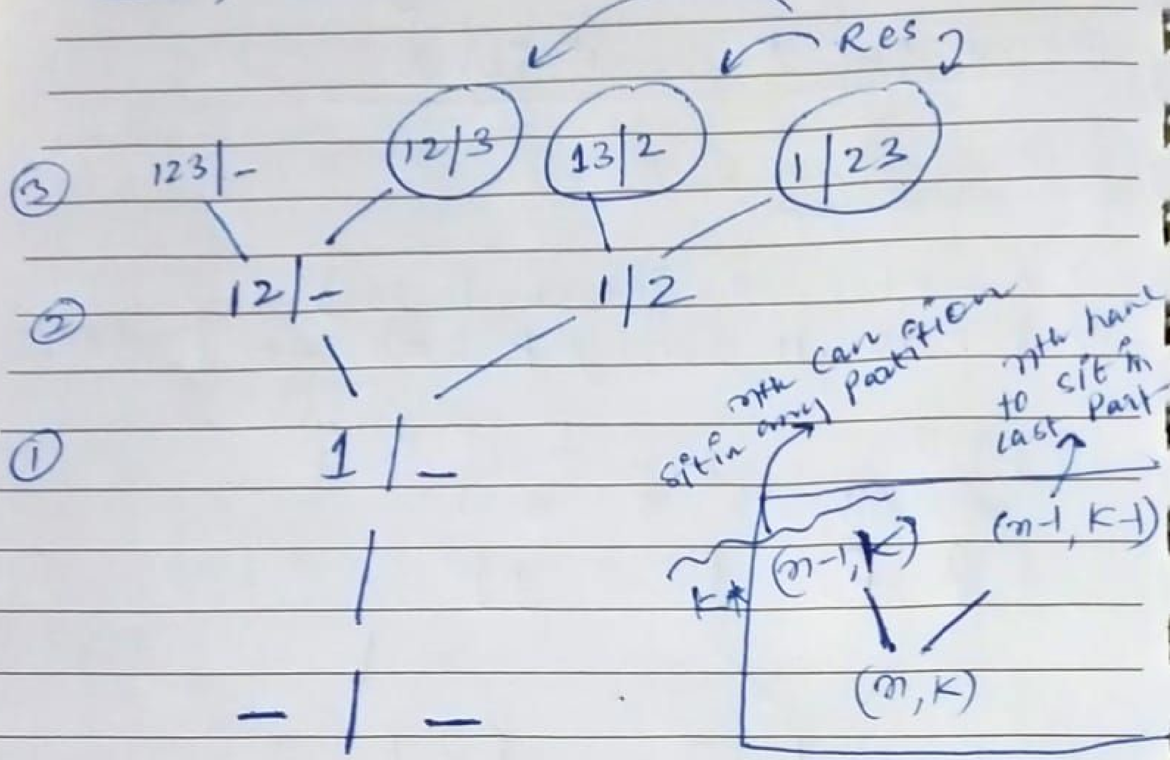
① Count number of ways to partion a set into K subsets → & PRint those subset

n = 3, K = 2



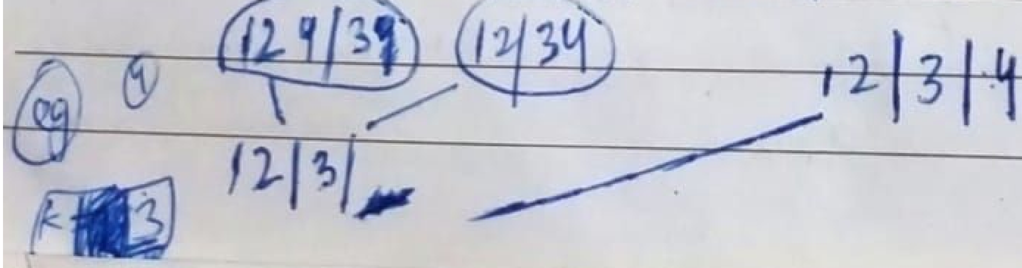
At each step add in existing as well as one more empty space

FOR count : $K * S(n-1, K) + S(n-1, k-1)$

base case : $n=0 \;||\; K=0 \;||\; K>n \to 0$

$n=K \;||\; K=1 \to 1$

eg



② Partition array to k-subsets (equal sum)

eg: {2, 1, 4, 5, 6} , k=3 } TRUE

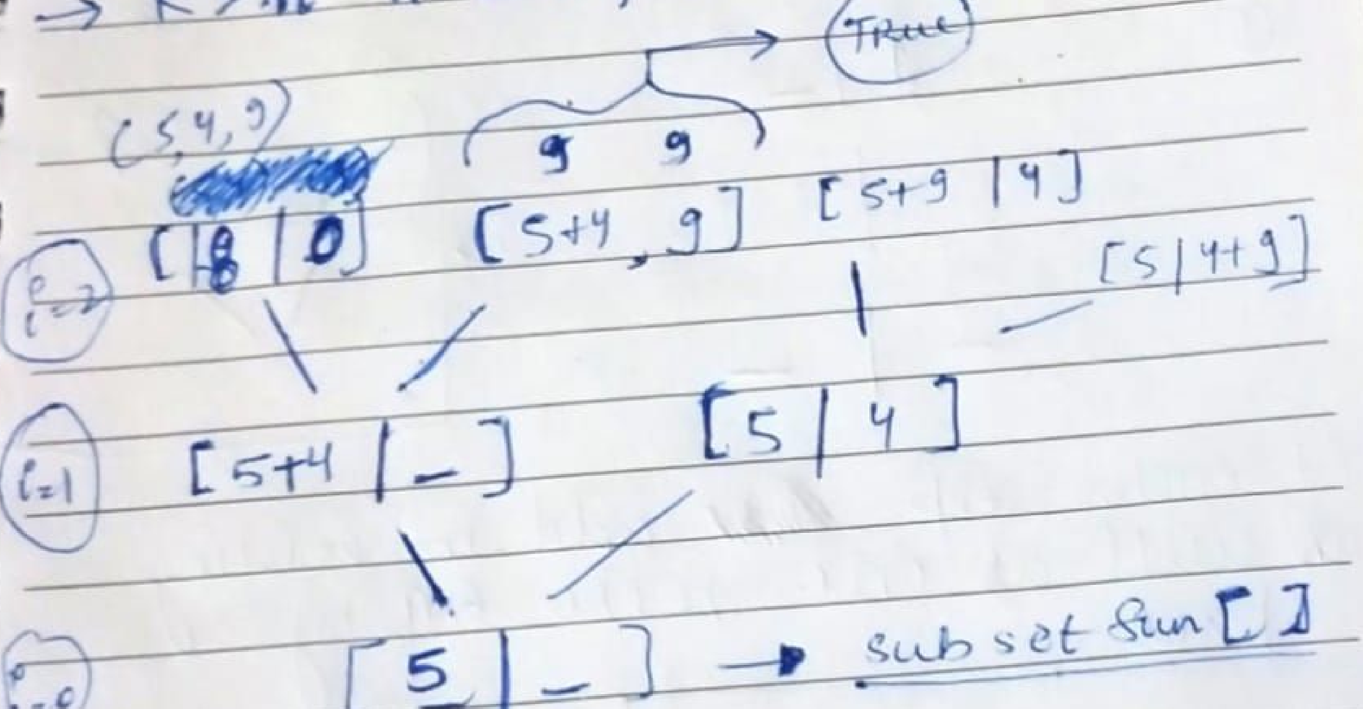
{2,4} {1,5} {6}

same as K partitions

Base cases :

→ k == 1 → true [arr]

→ K > n || sum % k != 0 → false



arr

{5, 4, 9} , k = 2

0 1 2

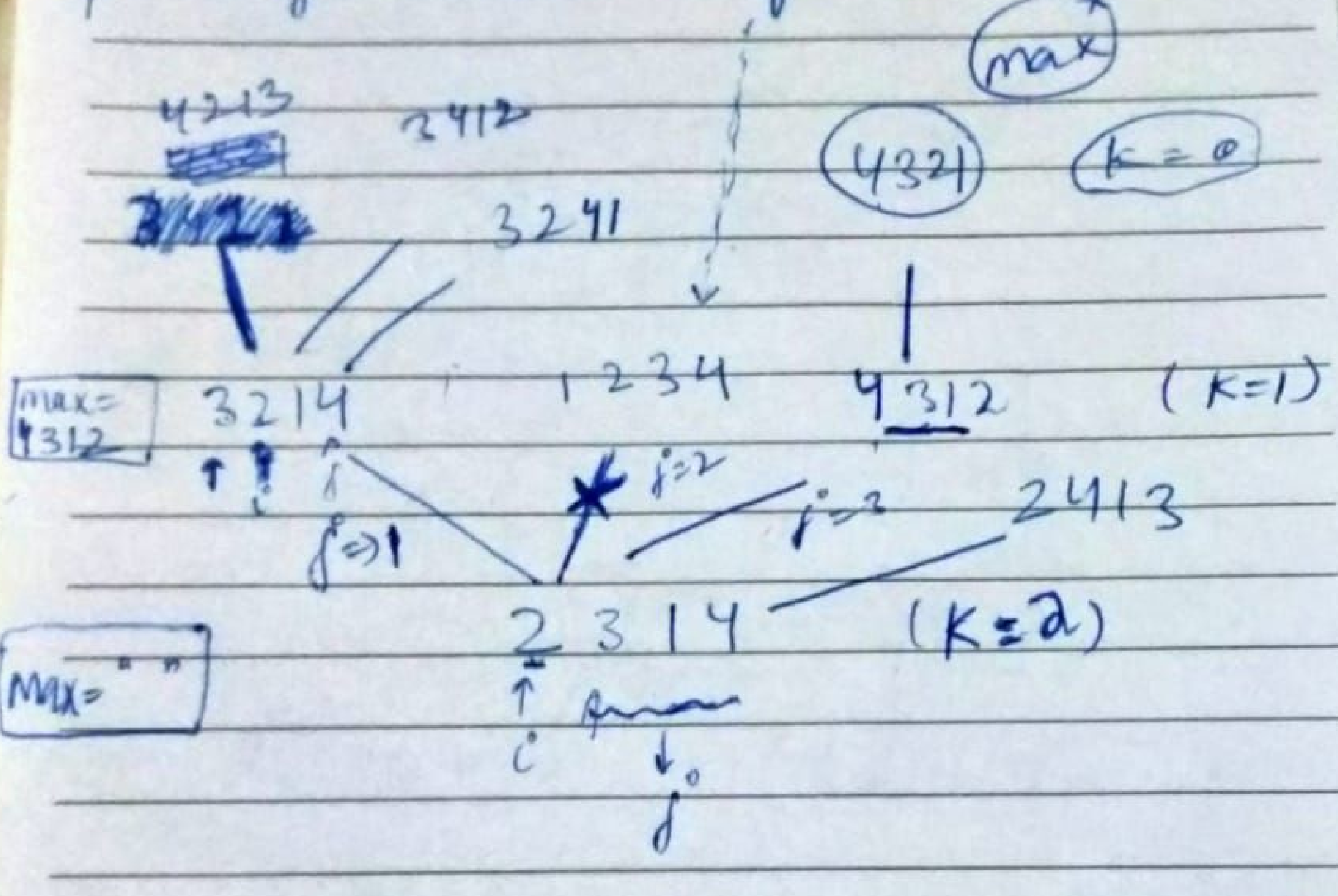
9 | 9 ✓

③ Maximum number formed After k - swaps

n => 2314 , k = 2

→ k=0 (return)

if arr[j] > arr[i] only then swap



TC(n^k)

SC(n)

At each step do:
i → 0 → < n-1
j = i+1 → < n }

(4) k th permultion of sequence

(eg) n=3, k→2 $\overset{123}{(132)}$ → $\boxed{\text{output}}$

Brute force → by Recursion store all Permutaions
& sort & tell the k th element
$O(n! \times n) + O(n \log n)$

Better Approach $TC(n^2)$ $SC(n)$ [3 4 1 2]

n=4, k=17 | 16 th   a = {1, 2, 3, 4} 16/6 → 2

① n!/n → 6 (groups) ② add a[k/groups]

① n=3, k = 16%6 → 4 (k % group), a = {1, 2, 4}
(n!)/n → 6/3 → ② groups
add a[4/2] → a[2] → 4

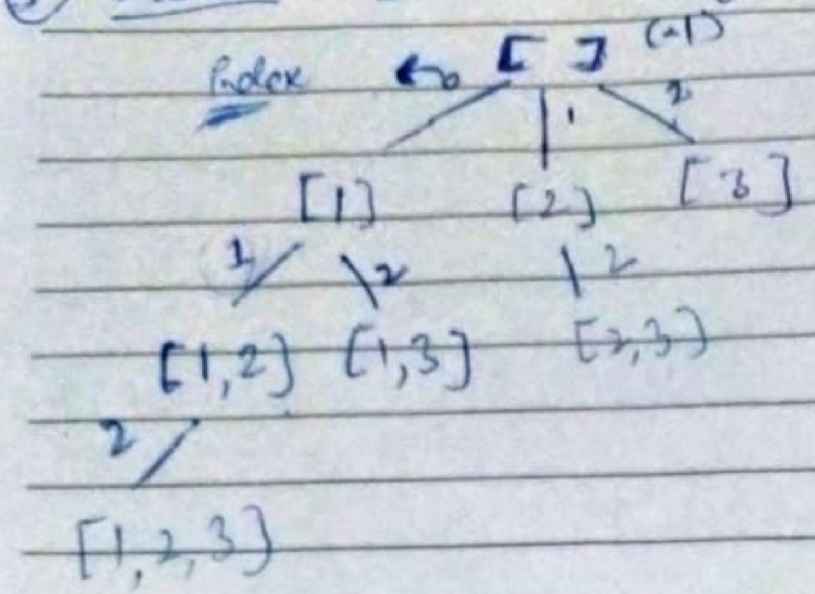
① n=2, k = 4%2 → 0, a = {1, 2}
→ groups → 2/2 → 1
→ add a[0/1] → a[0] → 1

① n=1, k → 0%1 → 0 a → {2}

Base case : n=1 simply add a[0]
return;

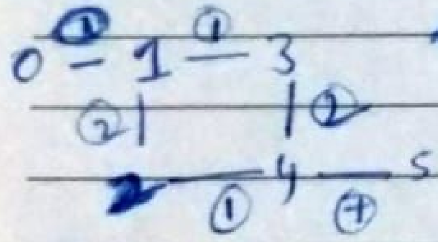
⑤ Subsets of An Array :

[1,2,3]



Path of greater than equal to K Length

E = 6 K = 10 V = 6



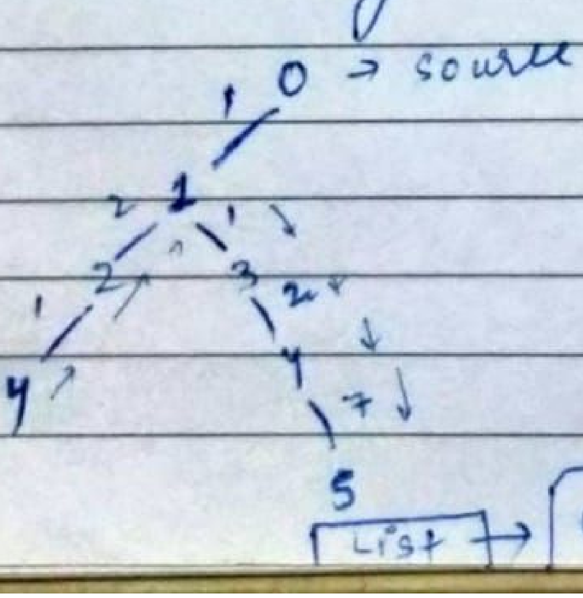
Path :

0-1-3-4-5

→ output

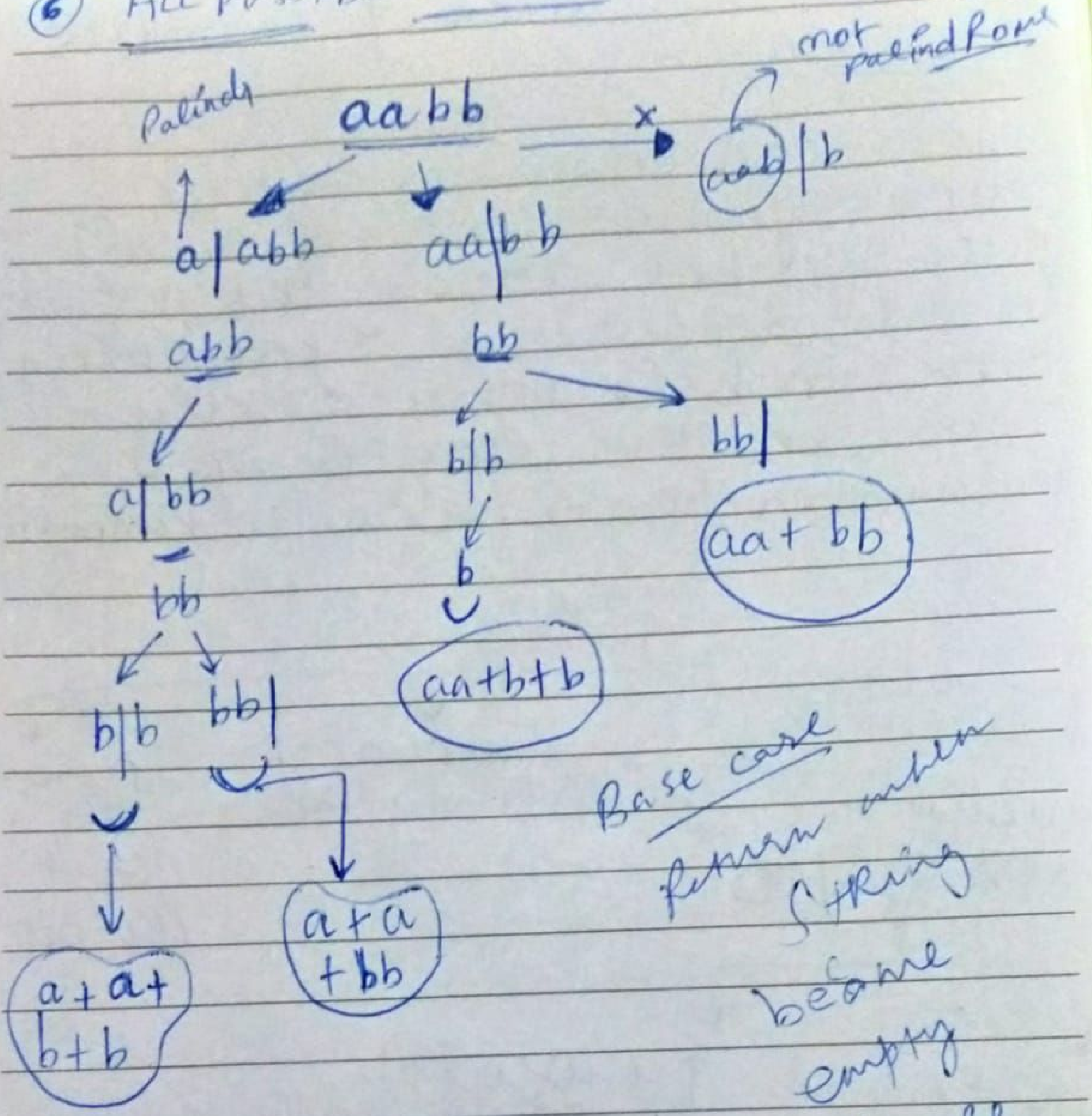
BackTracking is used as BFS/DFS not grab the path correctly.

{ BackTracking help to Track all the paths }



List → 0 → 1 → 3 → 4 → 5

(6) ALL POSSIBLE PaLindRomic PaRtitioning



Base case return when String become empty & store the tracked List

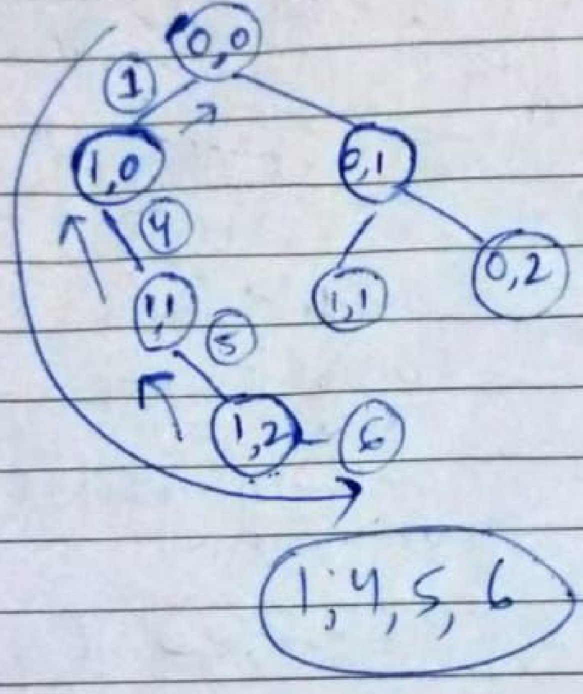
(7) PRint all Possible Paths from ..../......./2020

top-Left to bottom-right → (↓ →)

↓ Possible DiR

eg $\begin{Bmatrix}1&2&3\\4&5&6\end{Bmatrix} \rightarrow \begin{Bmatrix}1&2&3&6\\1&4&5&6\\1&2&5&6\end{Bmatrix}$

Input output



Base case

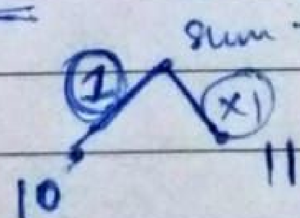
when Reach at Last cell

display The TRack

(8) Partition equal subset sum :

can partition in 2 subset oR not →

eg [1,5,11,5] → [1,5,5] [11]

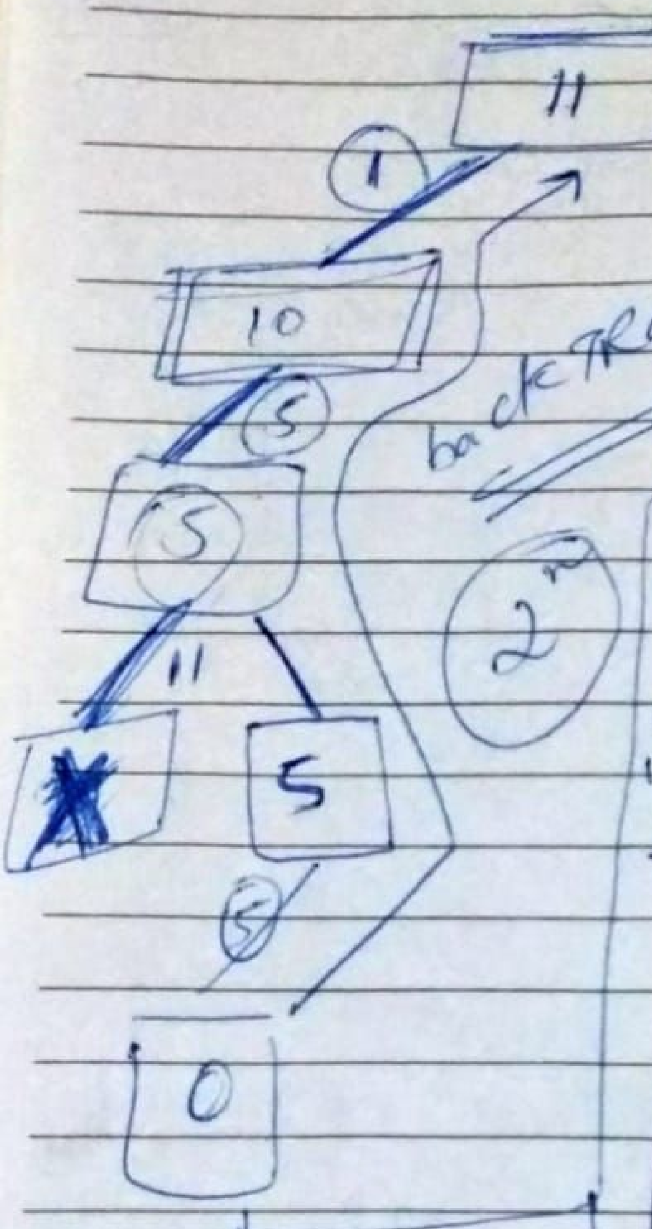


} simple Find sum in ARray

subsetsum Problem

8 All sums with given Sum :

(Subset sum Problem) -



elements not Repeated

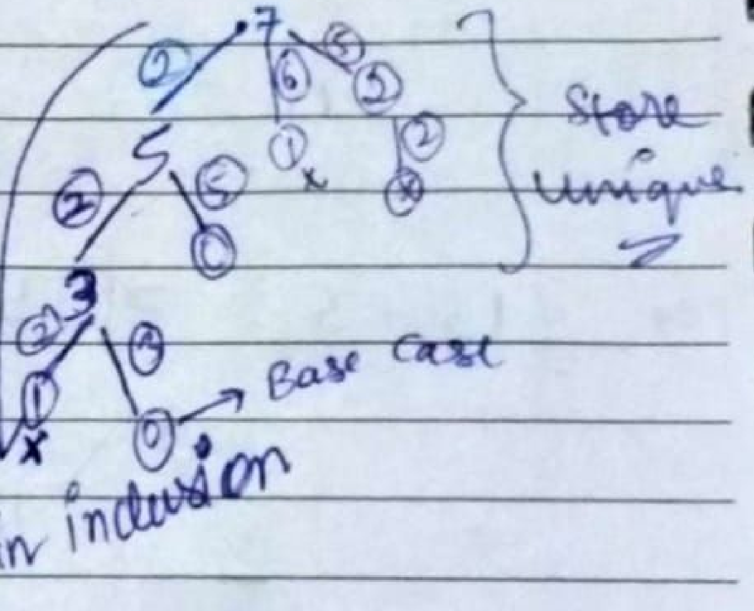
Combinational Sum --

candidates = [2,3,6,7,5]

target => 7

All combination with sum

→ [2,2,3] [7] [2,5]



↓

elements Repeated

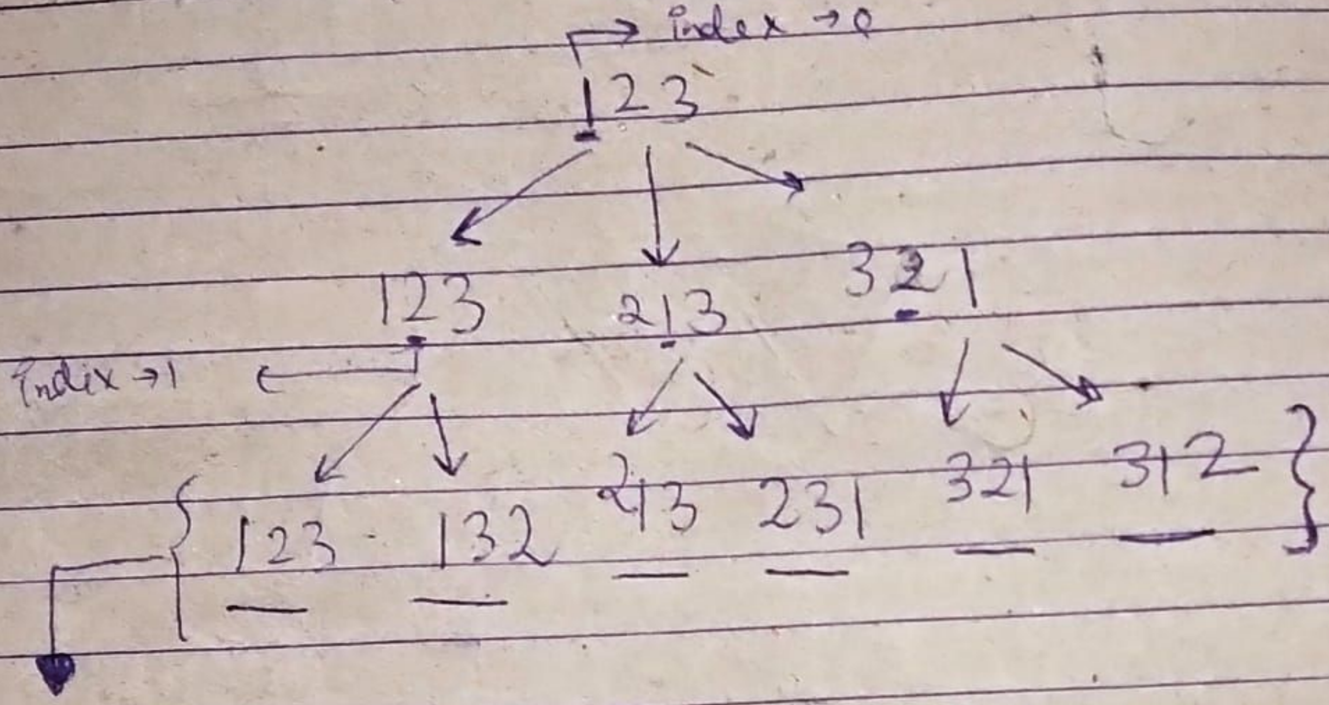
Permutations of string

BACKTRACKING

"123" →

123	321
132	312
213	231

→ output



Output

Base case index == arr.Length
Add into ans

Backtrack to
swap again