$n \leq 3$

$2 \leq 3$

↓

yes

↳ return $n = 2$

Getways (4) = Getways (3) + Getways (2)

$= 3 + 2$

$= 5$

Q Print all the sequences of a string using recursion.

Eg- {1, 2}

↓

{1}, {2}, {1, 2}

Eg- {1, 2, 3}

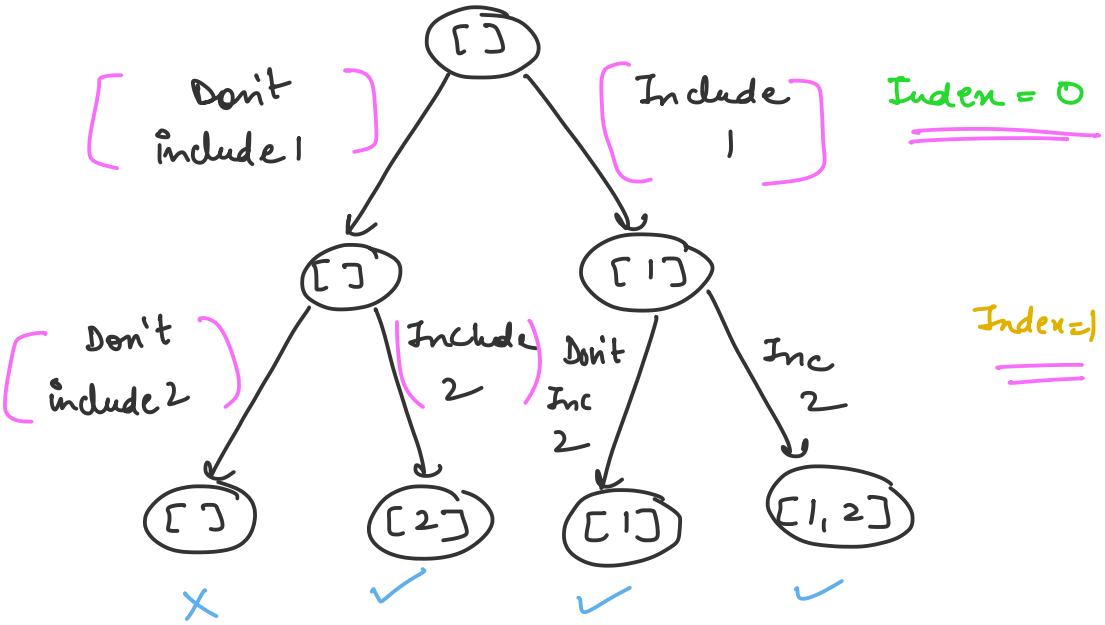
↓

{1}, {2}, {3}, {1, 2},

{1, 3}, {2, 3}, {1, 2, 3}

0 1

Eg - {1, 2}

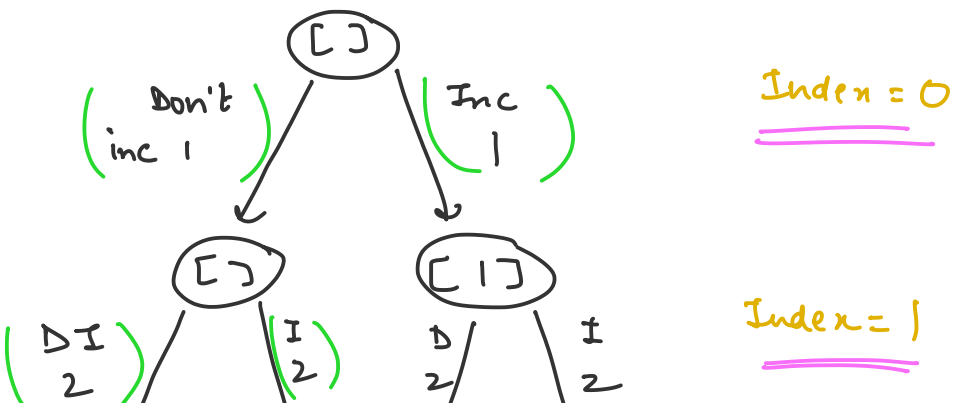


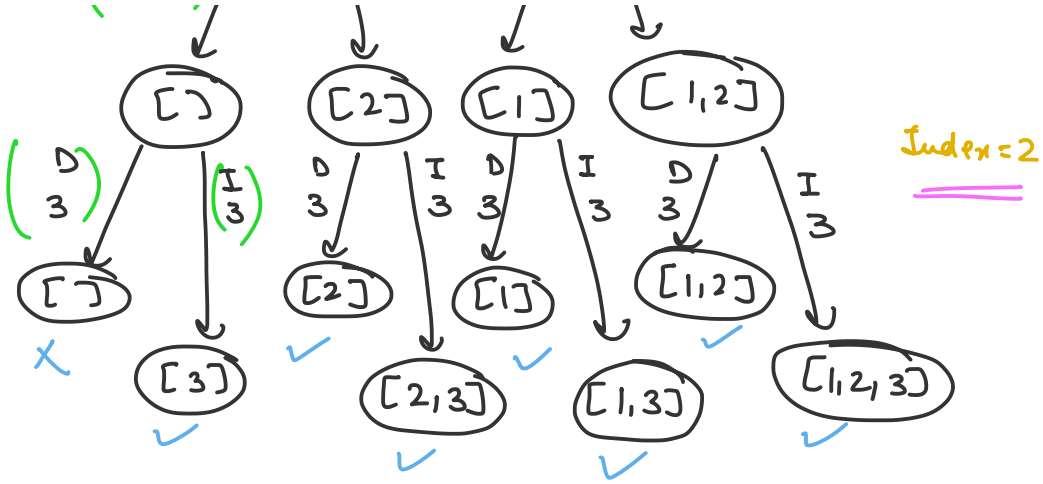
output = [2], [1], [1,2]

Another eg -

0 1 2

{1, 2, 3}





output - [3], [2], [2,3], [1], [1,3], [1,2], [1,2,3]

for every element in the array, we have 2 choices -

1. To include in the subsequence
2. To not include in the subsequence.

Apply this on all the elements in the array starting with index 0, and do this recursively until we reach the last index.

Base case -

if (index == length of array)

↓

print all the subsequences.

Recursive call

PS(arr, index, tempArr)

↳ To store subsequences

{

PS(arr, index+1, tempArr) → Include

Store in tempArr ← Add the value in tempArr

↓

arr [index]

PS (arr, index+1, tempArr) → Don't include

Remove the last value

from tempArr

}

Note -

Here tempArr is dynamic array - which is called as array list, because arrays are of fixed length, means they cannot grow or shrink in size. But arraylists can grow or shrink according to the requirement.

Dry Run -

```
        0  1
arr : {1, 2}
```

① PS (arr, 0, tempArr)

↓

tempArr = []

index = 0

arr.length = 2

if (0 == 2)

↓ no

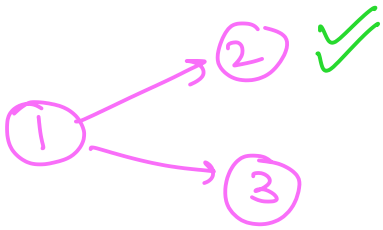
fails

② PS (arr, 1, tempArr);

tempArr.add (1)

③ PS (arr, 1, tempArr);

Remove last value from tempArr.



② PS (arr, 1, tempArr)

tempArr = []

index = 1

if (1 == 2)

↓ no

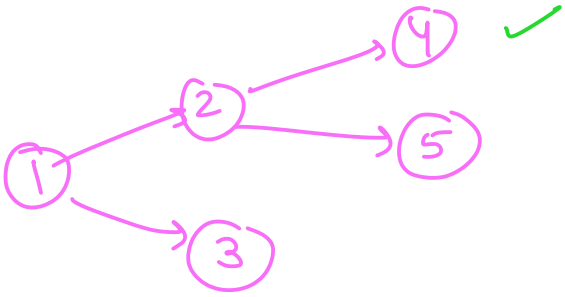
fails

④ PS (arr, 2, tempArr) ✓

tempArr.add (2) 2

(5) PS (arr, 2, tempArr)

Remove the last value from tempArr



(4) PS (arr, 2, tempArr)

tempArr = []

index = 2

if (2 == 2)

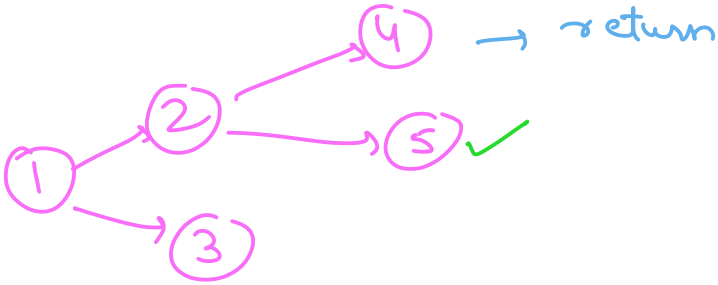
↓ yes

print

tempArr.size() = 0

↓

Return



(5) PS (arr, 2, tempArr)

↓

tempArr [2]

index = 2

if (2 == 2)

↓ yes

print → [2]

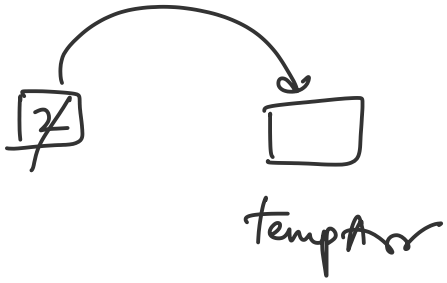
Remove the last value from tempArr

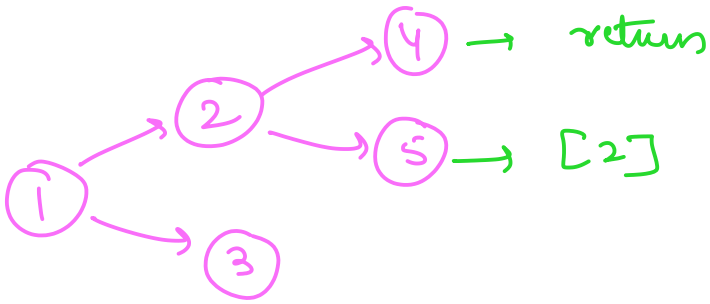
tempArr [2]

tempArr.remove (tempArr.size()-1)

tempArr.remove (1-1)

tempArr.remove (0)





② PS (arr, 1, tempArr); ✓

tempArr.add(1) [1] tempArr

③ PS (arr, 1, tempArr); ✓

Remove last value from tempArr.

③ PS (arr, 1, tempArr)

↓

tempArr = [1] ✓

index = 1

if (1 == 2)

↓ no

fails

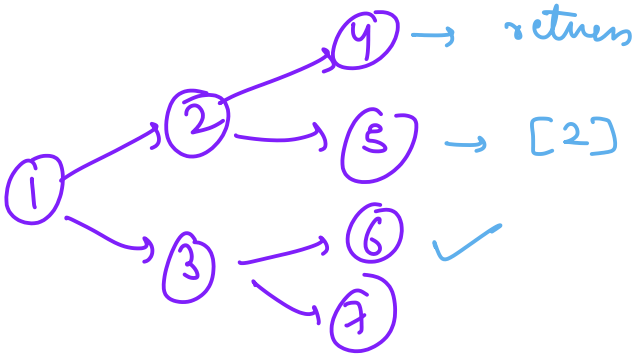
⑥ PS (arr, 2, tempArr) ✓✓

tempArr.add(2)

1	2

⑦ PS (arr, 2, tempArr) ✓✓

Remove last value



⑥ PS (arr, 2, tempArr)

tempArr [1]

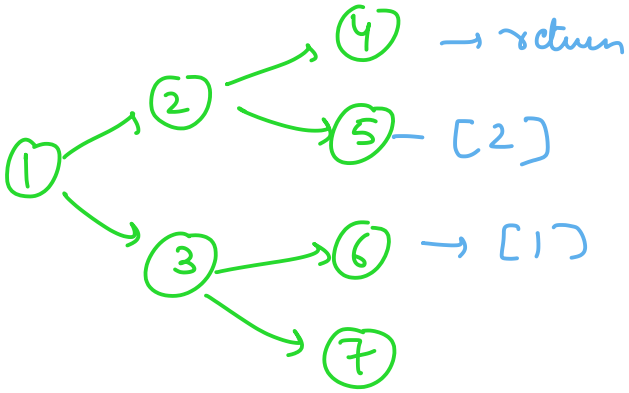
index = 2

if (index == arr.length)

if (2 == 2)

↓ yes

print [1]



(7) PS (arr, 2, tempArr)

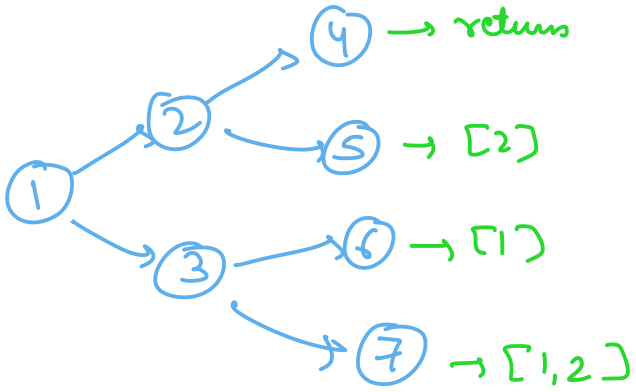
tempArr = [1, 2]

index = 2

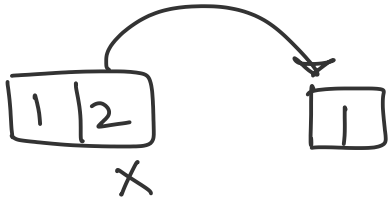
if (2 == 2)

↓ yes

print [1, 2]



Remove last value from tempArr



② PS (arr, 1, tempArr); ✓

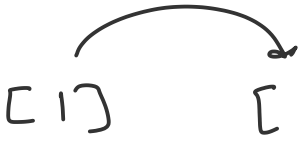
tempArr.add(1) ✓

③ PS (arr, 1, tempArr); ✓

Remove last value from tempArr.

↓

Remove last value



↓

Empty

Q Print all permutations of the given string

↓

Rearrangement of values

Eg-

string = "xy"