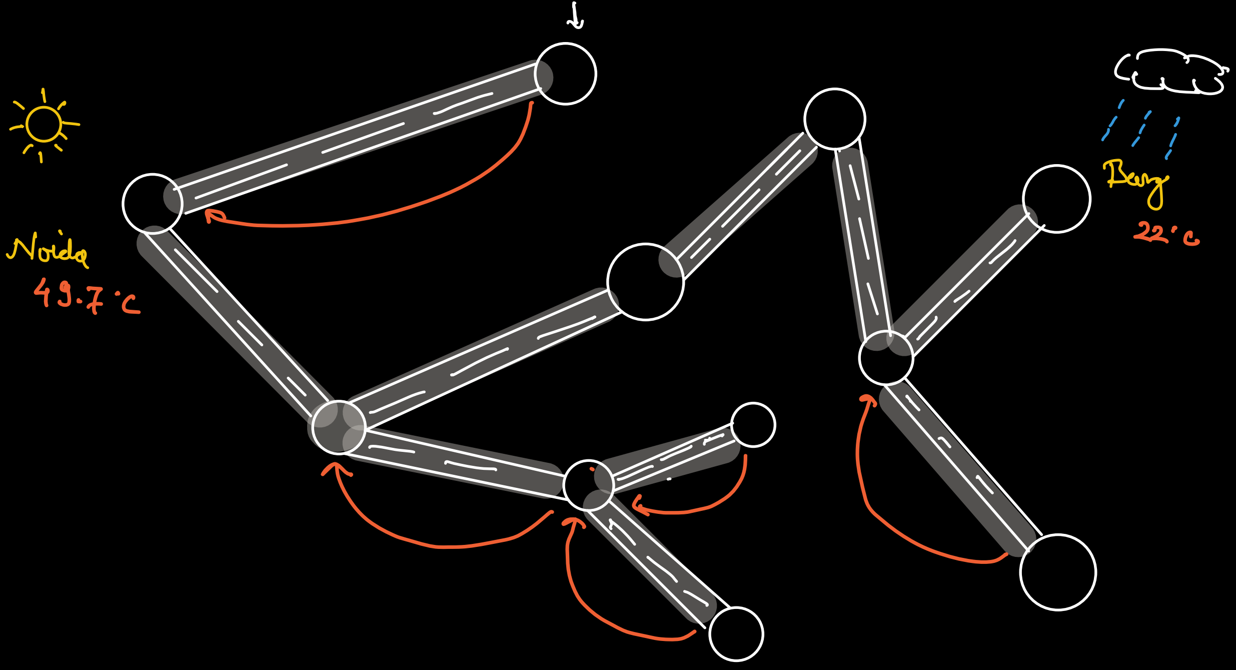
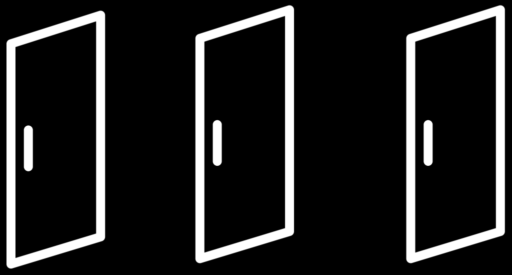




Pick a door

open & check

Close

Backtracking → Try out all possibilities (Brute force)

Q Print all N digit numbers using {1 & 2}

N = 1

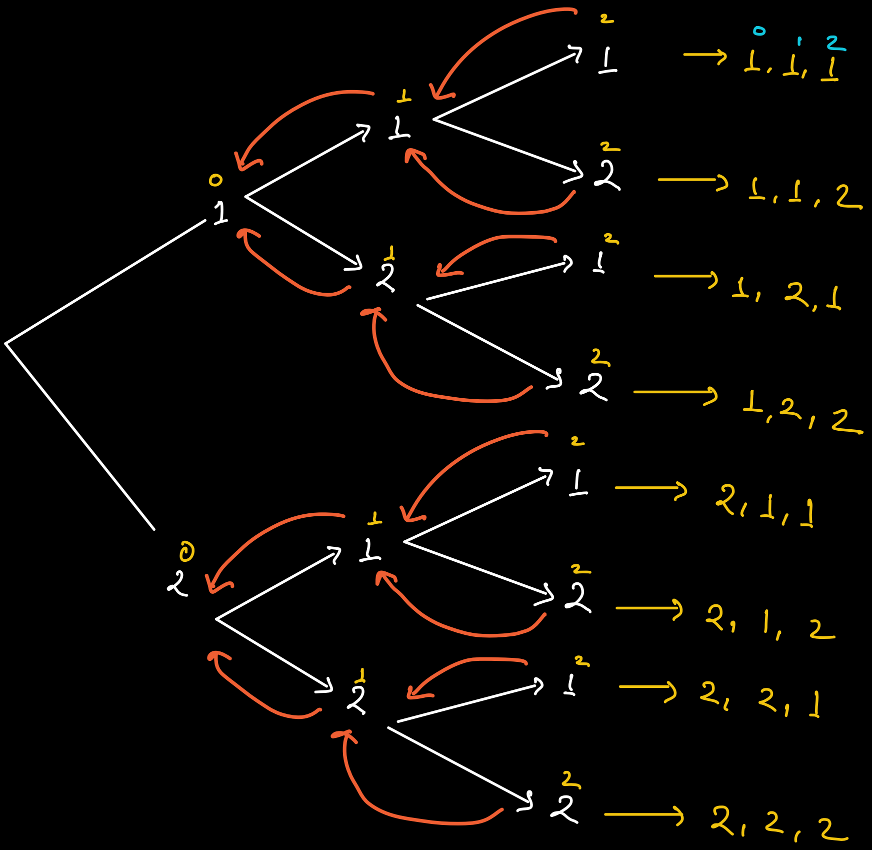
1
2

N = 2

1 1
1 2
2 1
2 2

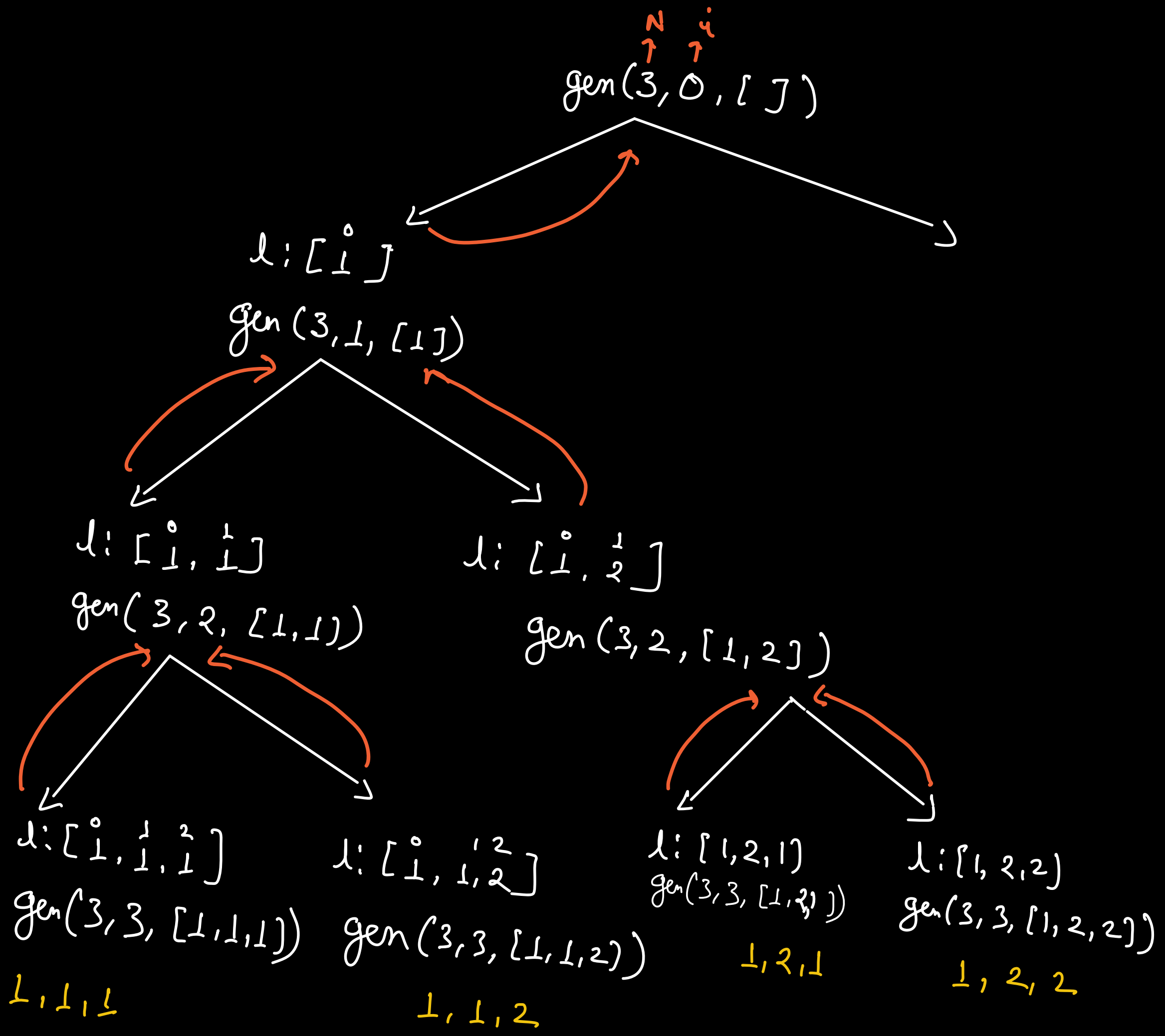
N = 3

[1, 1, 1]
[1, 1, 2]
[1, 2, 1]
[1, 2, 2]
[2, 1, 1]
[2, 1, 2]
[2, 2, 1]
[2, 2, 2]



```
                        0
                        ↑
void  generate ( N, index , curr list ) {
List< ArrayList <Int>>
      if (N == index) {
              Print(curr list); → O(N)
              ans ~~.add(curr list)~~;  → Clone & add
              ret;
      }
      curr list[index] = 1;
      generate(N, index+1, curr list);
      curr list[index] = 2;
      generate(N, index+1, curr list);
}
```

TC : O(N 2^N)
↓
Print



S copy / Referen

curr list

ans [[1,2,1] [1,2,1], [1,2,1]) . _

→ [[2,2,1]] . ___

Q Print all N digit no. (as lists) using {1, 2, 3, 4 & 5}

```
void generate ( N , index , currList ){

    if( N == index ){
        Print ( currList );
        ret;
    }

    currList[index] = 1;
    generate ( N , index+1, currList );

    currList[index] = 2 ;
    generate ( N , index+1, currList );

    currList[index] = 3 ;
    generate ( N , index+1, currList );

    currList[index] = 4 ;
    generate ( N , index+1, currList );

    currList[index] = 5 ;
    generate ( N , index+1, currList );
}
```

```
for( i = 1; i <= 5; i++ ){
    currList [index ] = i;
    generate (N, index +1 , currList);
}
```

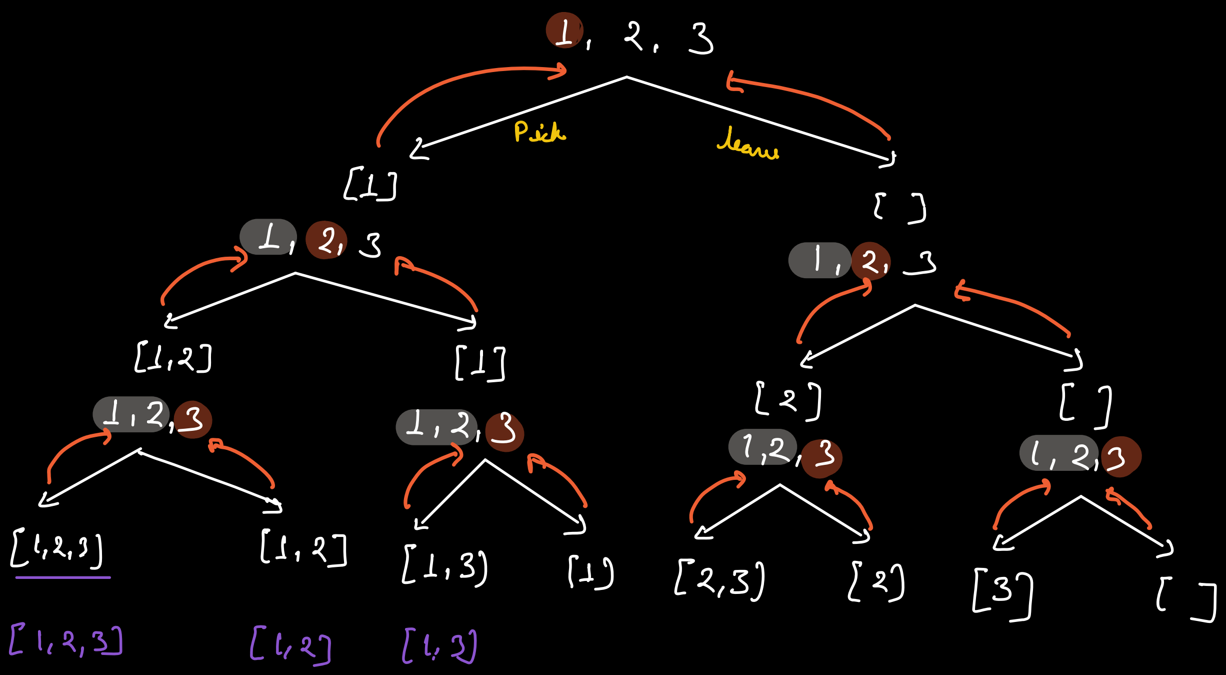
TC: $O(N 5^N)$

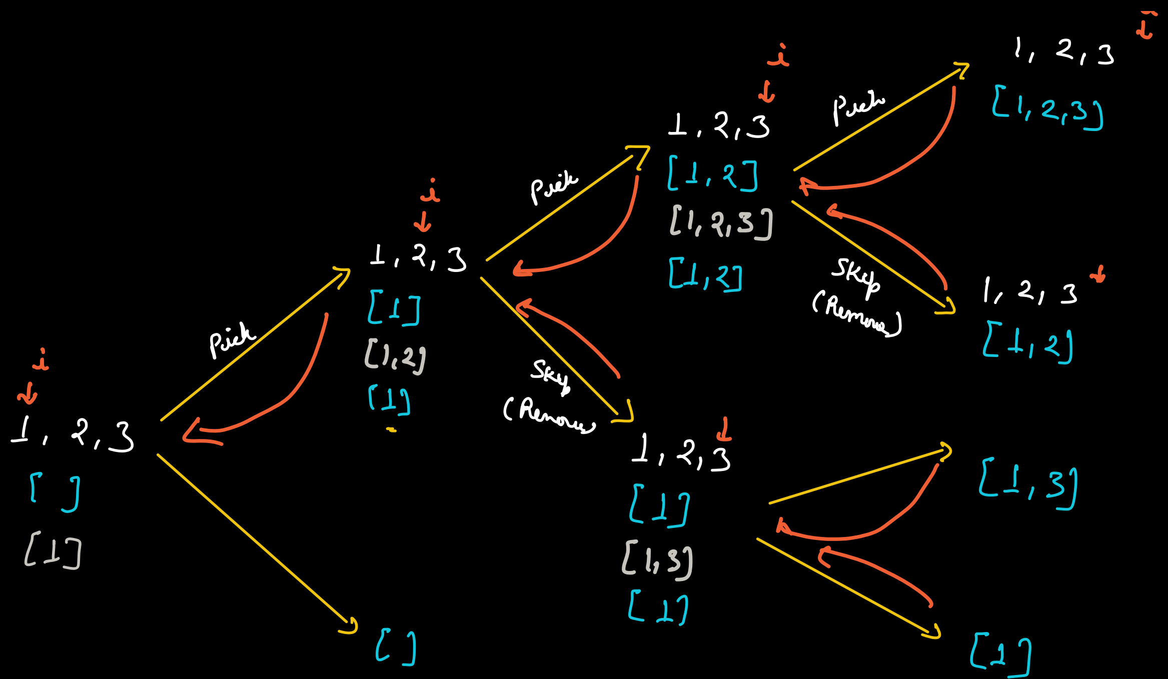
Amazon
MS
Facebook
⋮

Q Given an array. Generate all subsets of it.

A : [1, 2, 3]

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





```
getAllSubsets(CurrList< >, index, A[], List<List<int>> ans){
    if (index == A.size()){
        ans.add(deep copy/clone of currList);
        ret;                                      O(N)
    }
    CurrList.add(A[index]);
    getAllSubsets(currList, index+1, A, ans);
    CurrList.pop(); //delete the last added element
    getAllSubsets(currList, index+1, A, ans);
}
```

TC : $O(N\,2^N)$

SC : $O(N)$ + $O(N)$ ⇒ $O(N)$ (Excluding output)

Recursion stack + Curr List

Intuit
Amazon
Flipkart
Ola
MS

Q Count the no. of ~~subsets~~ (subsq) with sum = K.

→ No duplicats

A : [5, 2, 7]   K = 7

{5, 2}
{7}
→ 2

A : [2, 2, 4, 4]   K = 6

{5, 2}
{2, }

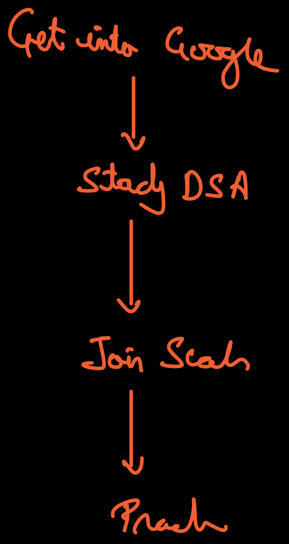
```
int   getAllSubsets( A[], K, index, curSum ){
          if ( index == A.length ){
                 if ( curSum == K){
                       ret 1;
                 }
                 else
                       ret 0;
          }

      curSum = curSum + A[index];
      x = getAllSubsets(A, K, index+1, curSum);
      curSum = curSum - A[index];
     y = getAllSubsets(A, K, index+1, curSum);
     ret x+y;
  }
```

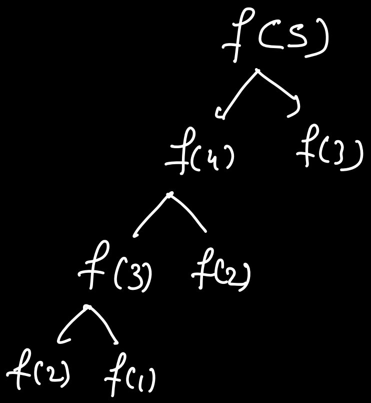
Q Given an array. Print all the permutations of the array.
└→ (without duplicates)

A : 1, 2, 4

0	1	2
1	2	4
1	4	2
2	1	4
2	4	1
4	1	2
4	2	1

⟶ N!





0	1	2	3	4	5
0	1	1	2	3	5