75 Days of Code
Day 58
Problem no : Leetcode 216
Problem Title :. Combination Sum III
Problem type : recursion

Find all valid combinations of k numbers that sum up to n such that the following conditions are true:

Only numbers 1 through 9 are used.
Each number is used at most once.
Return a list of all possible valid combinations. The list must not contain the same combination twice, and the combinations may be returned in any order.

Example 1:

Input: k = 3, n = 7
Output: [[1,2,4]]
Explanation:
1 + 2 + 4 = 7
There are no other valid combinations.
Example 2:

Input: k = 3, n = 9
Output: [[1,2,6],[1,3,5],[2,3,4]]
Explanation:
1 + 2 + 6 = 9
1 + 3 + 5 = 9
2 + 3 + 4 = 9

There are no other valid combinations.
Example 3:

Input: k = 4, n = 1
Output: []
Explanation: There are no valid combinations.
Using 4 different numbers in the range [1,9], the smallest sum we can get is 1+2+3+4 = 10 and since 10 > 1, there are no valid combination.

```typescript
function combinationSum3(k: number, n: number): number[][] {

    const result :number[][] =[];

    const recursiveCall =(start:number ,target:number,path:number[])=>{
        if(path.length===k && target===0){
            result.push([...path]);
            return ;
        }

        if(path.length>= k || target<=0){
            return;
        }

        for(let index = start ;index<=9 ;index++){
            path.push(index);
            recursiveCall(index+1 ,target-index,path);
            path.pop()

        }
    }

    recursiveCall(1,n,[]);
    return result;

};
```

Accepted

Editorial

Runtime	Details	Memory
45 ms		42.94 MB
Beats 97.22% of users with TypeScript		Beats 77.08% of users with TypeScript