Problem 216: Combination Sum III

Problem Information

Difficulty: Medium
Acceptance Rate: 72.63%
Paid Only: No
Tags: Array, Backtracking

Problem Description

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.

Constraints:

* `2 <= k <= 9` * `1 <= n <= 60`

Code Snippets

C++:

```
class Solution {
public:
vector<vector<int>> combinationSum3(int k, int n) {

}
};
```

Java:

```
class Solution {
public List<List<Integer>> combinationSum3(int k, int n) {

}
}
```

Python3:

```
class Solution:
def combinationSum3(self, k: int, n: int) -> List[List[int]]:
```