Problem 40: Combination Sum II

Problem Information

Difficulty: Medium
Acceptance Rate: 58.49%
Paid Only: No
Tags: Array, Backtracking

Problem Description

Given a collection of candidate numbers (`candidates`) and a target number (`target`), find all unique combinations in `candidates` where the candidate numbers sum to `target`.

Each number in `candidates` may only be used **once** in the combination.

Note: The solution set must not contain duplicate combinations.

Example 1:

Input: candidates = [10,1,2,7,6,1,5], target = 8 **Output:** [[1,1,6], [1,2,5], [1,7], [2,6]]

Example 2:

Input: candidates = [2,5,2,1,2], target = 5 **Output:** [[1,2,2], [5]]

Constraints:

* `1 <= candidates.length <= 100` * `1 <= candidates[i] <= 50` * `1 <= target <= 30`

Code Snippets

C++:

```
class Solution {
public:
```

```
vector<vector<int>> combinationSum2(vector<int>& candidates, int target) {

}
};
```

Java:

```
class Solution {
public List<List<Integer>> combinationSum2(int[] candidates, int target) {

}
}
```

Python3:

```
class Solution:
def combinationSum2(self, candidates: List[int], target: int) ->
List[List[int]]:
```