40. Combination Sum II

Medium 1933 71 Add to List Share

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

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

Note:

- All numbers (including `target`) will be positive integers.
- The solution set must not contain duplicate combinations.

Example 1:

```
Input: candidates = [10,1,2,7,6,1,5], target = 8,
A solution set is:
[
  [1, 7],
  [1, 2, 5],
  [2, 6],
  [1, 1, 6]
]
```

Example 2:

```
Input: candidates = [2,5,2,1,2], target = 5,
A solution set is:
[
  [1,2,2],
  [5]
]
```

```java
class Solution {
    public List<List<Integer>> combinationSum2(int[] candidates, int target)
   {
        Arrays.sort(candidates);
         ArrayList<List<Integer>> res = new ArrayList<List<Integer>>();
         // no need to use loop here because do not want repetition of the
same number again and again.
         // like other problem in the same package from leetcode_39
         combinationSumHelper( candidates, 0, target, new ArrayList<Integer>
(), res);
         return res;
    }

    public static void combinationSumHelper( int[] Allcandidates, int
startIdx, int target, List<Integer> curr,
              List<List<Integer>> res) {
         // if (target < 0) return; // this also work instead of
'Allcandidates[i] <= target'
         if (target == 0) {
             res.add(curr);
             return;
         }
         for (int i = startIdx; i < Allcandidates.length && Allcandidates[i]
<= target; i++) {
```

Testcase Run Code Result Debugger

Accepted Runtime: 0 ms

Your input
```
[10,1,2,7,6,1,5]
8
```

Output `[[1,1,6],[1,2,5],[1,7],[2,6]]` Diff

Expected `[[1,1,6],[1,2,5],[1,7],[2,6]]`