Combinations Sum II

Question: Given a collection of candidate numbers (C) and a target number (T), find all unique combinations in C where the candidate numbers sums to T.

Each number in C may only be used once in the combination.

Note:

All numbers (including target) will be positive integers.

Elements in a combination (a1, a2, ⋯ , ak) must be in non-descending order. (ie, a1 ⩽ a2 ⩽ ⋯ ⩽ ak).

The solution set must not contain duplicate combinations.

For example, given candidate set 10,1,2,7,6,1,5 and target 8,

A solution set is: [1, 7] ; [1, 2, 5] ; [2, 6] ; [1, 1, 6]

Solutions:

```
class Solution:
    def combinationSum2(self, candidates, target):
        if not candidates:
            return []
        candidates.sort()
        result = []
        self.combination(candidates, target, [], result)
        return result

    def combination(self, candidates, target, current, result):
        s = sum(current) if current else 0
        if s > target:
            return
        elif s == target:
```

```
            result.append(current)
            return
        else:
            i = 0
            while i < len(candidates):
                self.combination(candidates[i + 1:], target, current + [candidates[i]],
result)
                while i + 1 < len(candidates) and candidates[i] == candidates[i + 1]:
                    i += 1
                i += 1


Solution().combinationSum2([10, 1, 2, 7, 6, 1, 5], 8)
```