213. 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. 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]
]

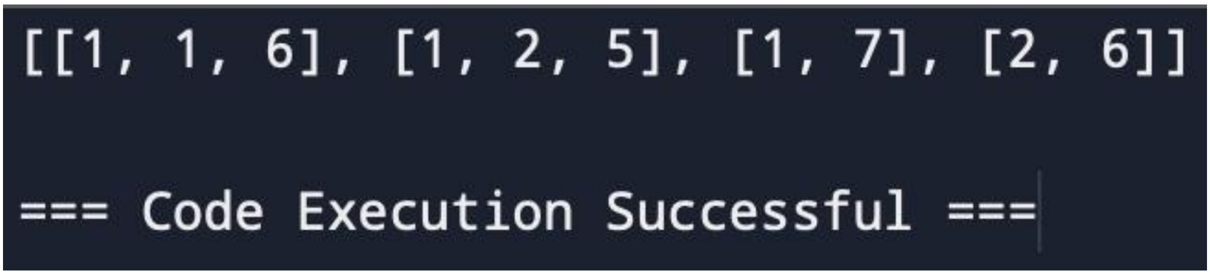
PROGRAM:-

```
def combinationSum2(candidates, target):
    def backtrack(start, target, path):
        if target == 0:
            res.append(path)
            return
        for i in range(start, len(candidates)):
            if i > start and candidates[i] == candidates[i-1]:
                continue
            if candidates[i] > target:
                break
            backtrack(i+1, target-candidates[i], path + [candidates[i]])

    candidates.sort()
    res = []
    backtrack(0, target, [])
    return res

# Example
candidates = [10, 1, 2, 7, 6, 1, 5]
target = 8
print(combinationSum2(candidates, target))
```

OUTPUT:-



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

=== Code Execution Successful ===
```

TIME COMPLEXITY:-O(n log n)