

68. 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]]`

AIM: To find the combination Sum

PROGRAM:

```
def combinationSum2(candidates, target):
    def backtrack(start, target, path):
        if target == 0:
            result.append(path[:])
            return
        if target < 0:
            return
        for i in range(start, len(candidates)):
            if i > start and candidates[i] == candidates[i - 1]:
                continue
            path.append(candidates[i])
            backtrack(i + 1, target - candidates[i], path)
            path.pop()
        result = []
        candidates.sort()
        backtrack(0, target, [])
        return result
    candidates = [10, 1, 2, 7, 6, 1, 5]
    target = 8
    print(combinationSum2(candidates, target))
```

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

OUTPUT:

TIME COMPLEXITY: $O(n \log n + n^m)$