

```

1  /*
2  给定一个数组 candidates 和一个目标数 target ，找出 candidates 中所有可以使数字和为
target 的组合。
3
4  candidates 中的每个数字在每个组合中只能使用一次。
5
6  说明：
7
8      所有数字（包括目标数）都是正整数。
9      解集不能包含重复的组合。
10
11  示例 1：
12
13  输入：candidates = [10,1,2,7,6,1,5]，target = 8，
14  所求解集为：
15  [
16      [1, 7]，
17      [1, 2, 5]，
18      [2, 6]，
19      [1, 1, 6]
20  ]
21
22  示例 2：
23
24  输入：candidates = [2,5,2,1,2]，target = 5，
25  所求解集为：
26  [
27      [1,2,2]，
28      [5]
29  ]
30
31  来源：力扣（LeetCode）
32  链接：https://leetcode-cn.com/problems/combination-sum-ii
33  著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
34  */

```

分析:

- 与39题类似，需要注意的是，要剪枝重复的部分，需要排序以及进行访问标记。

方法一:C++_回溯剪枝

```

1  class Solution
2  {
3
4      private:
5
6      void helper(      vector<vector<int>>&      vvi      ,

```

```

7         vector<int>&      candidates      ,
8         int              target          ,
9         int              cur_sum         ,
10        vector<int>&      cur_vi          ,
11        int              idx             ,
12        vector<int>&      visited
13    )
14    {
15        if(cur_sum == target)
16        {
17            vvi.push_back(cur_vi);
18            return ;
19        }
20
21
22
23        for(int i = idx ; i < candidates.size() ; i++)
24        {
25            if(cur_sum+candidates[i] > target)
26            {
27                continue;
28            }
29
30            if(i>0 && candidates[i] == candidates[i-1] && !visited[i-
1])
31            {
32                continue;
33            }
34
35            cur_vi.push_back(candidates[i]);
36            visited[i] = 1;
37
38            helper(vvi,candidates,target,cur_sum+candidates[i],cur_vi,i+1,visited);
39            visited[i] = 0;
40            cur_vi.pop_back();
41        }
42    }
43
44
45    public:
46    vector<vector<int>> combinationSum2(vector<int>& candidates, int
target)
47    {
48        vector<vector<int>> vvi;
49
50        if(candidates.size() < 1)
51        {
52            return vvi;
53        }
54        vector<int> cur_vi;
55        vector<int> visited(candidates.size(),0);
56        sort(candidates.begin(),candidates.end());
57        helper(vvi,candidates,target,0,cur_vi,0,visited);
58        return vvi;
59
60
61    }

```

```
62  };
63
64
65  /*
66  执行结果:
67  通过
68  显示详情
69  执行用时 :12 ms, 在所有 cpp 提交中击败了79.44% 的用户
70  内存消耗 :9.1 MB, 在所有 cpp 提交中击败了90.14%的用户
71  */
```

AlimyBreak
2019.11.29