40. Combination Sum II

Medium 629 34 Fav

Given a collection of candidate number
combinations in `candidates` where th

Each number in `candidates` may only

Note:

- All numbers (including `target`)
- The solution set must not contai

Example 1:

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

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

Example 2:

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

```java
package Algorithm;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

/*
 * 这道题目还是利用回溯递归的方法
 */
public class L40 {
    public List<List<Integer>> combinationSum2(int[] candidates, int target) {
        List<List<Integer>> res = new ArrayList<List<Integer>>();
        if(candidates == null || candidates.length == 0 || target < 0)
            return res;
        List<Integer> list = new ArrayList<Integer>();
        Arrays.sort(candidates);//因为有重复的数字，所以需要进行排序去重
        get(candidates, res, list, target, 0);
        return res;
    }

    private void get(int [] candidates, List<List<Integer>> res, List<Integer> list, int target, int i) {
        if(i > candidates.length || target < 0)
            return;
        if(target == 0) {
            res.add(new ArrayList<Integer>(list)); //这里如以前一样需要注意
            return;
        }
        for(int p = i; p < candidates.length; p ++) { //因为这道题目是可以下标不连续，所以需要从i到candidates的最后
            list.add(candidates[p]);  //添加
            get(candidates, res, list, target - candidates[p], p + 1);
            list.remove(list.size() - 1); //回溯删除
            while(p < candidates.length && candidates[p] == candidates[p + 1]) {
                p ++;//去除重复的数字
            }
        }

    }
}
```