040. Combination Sum II

040 Combination Sum II

- **BackTracking**+array

Description

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.
- 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]
]
```

1. Thought line

2. BackTracking+array

```
class Solution {
private:
    void backTrackingSum(vector<int>& nums, int target, int sum, int st, vector<vector<int>>& result, vector<int>& temp){
        if (sum == target) {
            result.push_back(temp);
            return;
        }
        if (st>=nums.size() || sum > target || sum+nums[st]>target) return;

        for (int i = st; i<=nums.size()-1; ++i){
            temp.push_back(nums[i]);
            backTrackingSum(nums, target, sum+nums[i], i+1, result, temp);
            temp.pop_back();
            /* avoid duplicate elements */
            while(i+1<=nums.size()-1&&nums[i+1]==nums[i]) ++i;
        }
    }

public:
    vector<vector<int>> combinationSum2(vector<int>& candidates, int target) {
        vector<vector<int>> result;
        vector<int> temp;
        sort(candidates.begin(),candidates.end());
        backTrackingSum(candidates, target, 0, 0, result, temp);
        return result;
    }
};
```