一、题目说明

题目39. Combination Sum，是从正数列表中选取几个，其和等于目标数的可能组合。任何一个数可以重复取，如candidates = [2,3,6,7], target = 7,结果集合是[[7], [2,2,3]]

如candidates = [2,3,5], target = 8,结果集合是 [[2,2,2,2], [2,3,3], [3,5]]

题目难度是Medium，先思考一下，再来解答。

二、我的解答

经过一番思考，这个可以画一个“树”，反映求解过程。这个图，我就不上了。我的代码：

```
#include<iostream>
#include<vector>
#include<algorithm>
using namespace std;
class Solution{
    public:
        void dfs(vector<vector<int>>& res,vector<int>& candidates,vector<int>&
path,int begin,int target){
            if(target==0){
                res.push_back(path);
                return;
            }
            for(int i=begin;i<candidates.size() && target-candidates[i]>=0;i++){
                path.push_back(candidates[i]);
                dfs(res,candidates,path,i,target-candidates[i]);
                path.pop_back();
            }
        }
        vector<vector<int>> combinationSum(vector<int>& candidates, int target){
            vector<vector<int>> res;
            vector<int> path;
            if(candidates.size()<1){
                return res;
            }
            sort(candidates.begin(),candidates.end());
            dfs(res,candidates,path,0,target);
            return res;
        }
};
int main(){
    Solution s;
    vector<int> candidates = {2,3,6,7};
    vector<vector<int>> res = s.combinationSum(candidates,7);
    cout<<"candidates of {2,3,6,7}"<<"\n";
    for(int i=0;i<res.size();i++){
        vector<int> r = res[i];
        for(int j=0;j<r.size();j++){
            cout<<r[j]<<" ";
        }
        cout<<"\n";
    }

    cout<<"candidates of {2,3,5}"<<"\n";
```

```
    candidates = {2,3,5};
    res = s.combinationSum(candidates,8);
    for(int i=0;i<res.size();i++){
        vector<int>r = res[i];
        for(int j=0;j<r.size();j++){
            cout<<r[j]<<" ";
        }
        cout<<"\n";
    }
    return 0;
}
```

性能：

```
Runtime: 12 ms, faster than 84.44% of C++ online submissions for Combination
Sum.
Memory Usage: 9.8 MB, less than 58.33% of C++ online submissions for Combination
Sum.
```

一行代码没修改，再次运行：

```
Runtime: 4 ms, faster than 99.94% of C++ online submissions for Combination Sum.
Memory Usage: 9.3 MB, less than 94.44% of C++ online submissions for Combination
Sum.
```

三、优化

比较搞笑的是，我一行代码没修改，再次提交性能居然大幅提高。厉害了！

另外的解答思路是DP，

用unordered_map<int, vector<vector>> dict;存储数的分解，比如求{2,3,6,7}和是8的：

dict[2] ={} {2}

dict[3] = {3}

dict[4] = {2,2}

dict[5] = dict[2] + dict[] {2,3}

dict[6] = {dict[2] + dict[2] + dict[2]},{dict[3]}

....

```
class Solution {
public:
    vector<vector<int>> combinationSum(vector<int> &candidates, int target)
    {
        unordered_map<int, vector<vector<int>>> dict;
        for (int i = 1; i <= target; i++)
            for (int it : candidates)
                if (i == it){
                    dict[i].push_back(vector<int>{ it });
                }
                else if (i > it){
                    for (auto ivec : dict[i - it]) {
```

```
                        if (it < ivec[ivec.size() - 1]) {
                            continue;
                        }
                        ivec.push_back(it);

                        dict[i].push_back(ivec);
                    }
                }

        return dict[target];
    }
};
```