

```

1  /*
2  给定一个可能包含重复元素的整数数组 nums，返回该数组所有可能的子集（幂集）。
3
4  说明：解集不能包含重复的子集。
5
6  示例：
7
8  输入：[1,2,2]
9  输出：
10 [
11     [2],
12     [1],
13     [1,2,2],
14     [2,2],
15     [1,2],
16     []
17 ]
18
19 在真实的面试中遇到过这道题？
20
21 来源：力扣（LeetCode）
22 链接：https://leetcode-cn.com/problems/subsets-ii
23 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。
24 */

```

分析:

- 方法一：暴力回溯法，然后利用sort和set去重,果断超时了.
- 方法二：回溯法，先对原始数组进行升序排序，在回溯过程中把数组分成两部分，一部分是已经访问过，一部分是未曾访问过，回溯过程中若当前索引对应的值与当前索引减一位置对应的值相等，且当前回溯位置比当层回溯的起始位置要大，则跳过该索引实现去重.

方法一:C++_set和sort去重.

```

1  class Solution
2  {
3
4      private:
5          void helper(      set<vector<int>>&    svi      ,
6                          vector<int>&        nums      ,
7                          vector<int>&        cur_vi     ,
8                          vector<int>&        visited    ,
9                          int                  idx
10                     )
11     {
12         if(idx == nums.size())
13         {
14             vector<int> temp(cur_vi);
15             sort(temp.begin(),temp.end());
16             svi.insert(temp);
17             return ;
18         }
19

```

```

20         for(int i = 0 ; i < nums.size();i++)
21         {
22             if(visited[i] == 1)
23             {
24                 continue;
25             }
26             else
27             {
28                 visited[i] = 1;
29                 helper(svi,nums,cur_vi,visited,idx+1);
30                 cur_vi.push_back(nums[i]);
31                 helper(svi,nums,cur_vi,visited,idx+1);
32                 cur_vi.pop_back();
33                 visited[i] = 0;
34             }
35         }
36     }
37
38     public:
39     vector<vector<int>> subsetsWithDup(vector<int>& nums)
40     {
41         set<vector<int>>      svi                ;
42         vector<vector<int>>    vvi                ;
43         vector<int>           cur_vi              ;
44         vector<int>           visited(nums.size(),0) ;
45
46
47         //sort(nums.begin(),nums.end());
48         helper(svi,nums,cur_vi,visited,0);
49
50         set<vector<int>>::iterator iter = svi.begin();
51         while(iter!=svi.end())
52         {
53             vvi.push_back(*iter);
54             iter++;
55         }
56         return vvi;
57     }
58 };
59 /*
60 16 / 19 个通过测试用例
61     状态：超出时间限制
62
63 提交时间：0 分钟之前
64 最后执行的输入： [1,2,3,4,5,6,7,8,10,0]
65 */

```

方法二:C++_排序回溯去重

```

1
2
3     class Solution
4     {
5     private:
6
7

```

```

8      void helper(    vector<vector<int>>&    vvi      ,
9                    vector<int>&            nums      ,
10                   vector<int>&            cur_vi     ,
11                   int                    idx
12                )
13            {
14                vvi.push_back(cur_vi); /* right 全不选依次保存*/
15
16                for(int i = idx ; i < nums.size() ; i++)
17                {
18                    /* idx的时候还是需要的*/
19                    if(    i > 0
20                        && nums[i] == nums[i-1]
21                        && i > idx
22                    )
23                    {
24                        continue;
25                    }
26                    cur_vi.push_back(nums[i]);
27                    helper(vvi,nums,cur_vi,i+1);
28                    cur_vi.pop_back();
29                }
30            }
31
32        public:
33            vector<vector<int>> subsetsWithDup(vector<int>& nums)
34            {
35
36                vector<vector<int>>    vvi;
37                if (nums.size() < 1)
38                {
39                    return vvi;
40                }
41                vector<int> cur_vi;
42                sort(nums.begin(), nums.end());
43                helper(vvi, nums, cur_vi, 0);
44                return vvi;
45            }
46        };
47
48        /*
49        执行结果:
50        通过
51        显示详情
52        执行用时 :12 ms, 在所有 cpp 提交中击败了82.24% 的用户
53        内存消耗 :9 MB, 在所有 cpp 提交中击败了97.02%的用户
54        */

```