

90. Subsets II

题目描述: <https://leetcode.com/problems/subsets-ii/>

给定一个数组（数组中可能有重复元素），求这个数组的所有可能的子集，要求子集不能有重复的，子集内的元素按照非降序排列。

例如：

```
[2,1,2]
```

返回：

```
[[],[1],[2],[1,2],[2,2],[1,2,2]]
```

解题思路：

如何避免重复？

记录下来每个重复元素的个数，则只考虑一次加入改值，在加入时，考虑加入的个数[1,sameNum]

代码：

```

class Solution {
public:
    vector<vector<int>> subsetsWithDup(vector<int>& nums) {
        sort(nums.begin(), nums.end());
        vector<vector<int> > res;
        res.push_back(vector<int>());
        for(int i = 0; i < nums.size();){
            int sameCount = 0;
            while(sameCount+i < nums.size() && nums[i] == nums[sameCount+i])
                sameCount++;
            int s = res.size();
            for(int j = 1; j <= sameCount; j++){
                vector<int> n;
                for(int k = 0; k < s; k++){
                    n = res[k];
                    for(int l = 0; l < j; l++){
                        n.push_back(nums[i]);
                    }
                    res.push_back(n);
                }
            }
            i+=sameCount;
        }
        return res;
    }
};

```