Permutations II

Question: Given a collection of numbers that might contain duplicates, return all possible unique permutations.

For example:

[1,1,2] have the following unique permutations:

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

Solutions:

```
class Solution:
    # @param num, a list of integer
    # @return a list of lists of integers
    def permuteUnique(self, num):
        length = len(num)
        if length == 0: return []
        if length == 1: return [num]
        num.sort()
        res = []
        previousNum = None
        for i in range(length):
            if num[i] == previousNum: continue
            previousNum = num[i]
            for j in self.permuteUnique(num[:i] + num[i+1:]):
                res.append([num[i]] + j)
        return res
Solution().permuteUnique([1,1,2])
```