3 Sum

Question: Given an array S of n integers, are there elements a, b, c in S such that a + b + c = 0? Find all unique triplets in the array which gives the sum of zero.

Note:

Elements in a triplet (a,b,c) must be in non-descending order. (ie, a $\leqslant$ b $\leqslant$ c)

The solution set must not contain duplicate triplets.

Solutions:

```
class Solution:
    # @return a list of lists of length 3, [[val1,val2,val3]]


    def threeSum(self, num):
        length = len(num)
        result = []


        if length < 3:
            return result


        num.sort()


        for i in range(length - 2):
            if i > 0 and num[i] == num[i - 1]:
                continue
            low = i + 1
            high = length - 1
            target_gap = 0 - num[i]
```

```
            while low < high:
                if num[low] + num[high] < target_gap:
                    low += 1
                    while low < high and num[low] == num[i - 1]:
                        low += 1
                elif num[low] + num[high] > target_gap:
                    high -= 1
                    while low < high and num[high] == num[high + 1]:
                        high -= 1
                else:
                    result.append([num[i], num[low], num[high]])
                    low += 1
                    while low < high and num[low] == num[low - 1]:
                        low += 1
                    high -= 1
                    while low < high and num[high] == num[high + 1]:
                        high -= 1
        return result

Solution().threeSum([-1, 0, 1, 2, -1, -4])
```