

N 数之和

LeetCode/[18. 四数之和](#)

题目描述（四数之和）

给定一个包含 n 个整数的数组 `nums` 和一个目标值 `target`，判断 `nums` 中是否存在四个元素 `a`, `b`, `c` 和 `d`，使得 $a + b + c + d$ 的值与 `target` 相等？找出所有满足条件且不重复的四元组。

注意：

答案中不可以包含重复的四元组。

示例：

给定数组 `nums = [1, 0, -1, 0, -2, 2]`，和 `target = 0`。

满足要求的四元组集合为：

```
[
  [-1, 0, 0, 1],
  [-2, -1, 1, 2],
  [-2, 0, 0, 2]
]
```

Python（N 数之和）

```
def nSum(A, N, t, tmp, ret):
    if len(A) < N or N < 2 or t < A[0] * N or t > A[-1] * N: # 结束条件
        return

    if N == 2:
        lo, hi = 0, len(A) - 1
        while lo < hi:
            s = A[lo] + A[hi]

            if s < t:
                lo += 1
            elif s > t:
                hi -= 1
            else:
                ret.append(tmp + [A[lo], A[hi]])
                lo += 1
                while lo < hi and A[lo] == A[lo - 1]: # 去重
                    lo += 1
    else:
        for i in range(len(A) - N + 1):
            if i > 0 and A[i] == A[i - 1]: # 去重
                continue
```

```
nSum(A[i+1:], N-1, t-A[i], tmp + [A[i]], ret)
```

```
class Solution:
```

```
    def fourSum(self, A, t):  
        """  
        :type A: List[int]  
        :type t: int  
        :rtype: List[List[int]]  
        """  
  
        A.sort()  
        ret = []  
  
        nSum(A, 4, t, [], ret)  
  
        return ret
```