# 446. Arithmetic Slices II - Subsequence

Given an integer array `nums`, return *the number of all the* ***arithmetic subsequences*** *of* `nums`.

A sequence of numbers is called arithmetic if it consists of **at least three elements** and if the difference between any two consecutive elements is the same.

- For example, `[1, 3, 5, 7, 9]`, `[7, 7, 7, 7]`, and `[3, -1, -5, -9]` are arithmetic sequences.
- For example, `[1, 1, 2, 5, 7]` is not an arithmetic sequence.

A **subsequence** of an array is a sequence that can be formed by removing some elements (possibly none) of the array.

- For example, `[2,5,10]` is a subsequence of `[1,2,1,2,4,1,5,10]`.

The test cases are generated so that the answer fits in **32-bit** integer.

**Example 1:**

```
Input: nums = [2,4,6,8,10]
Output: 7
Explanation: All arithmetic subsequence slices are:
[2,4,6]
[4,6,8]
[6,8,10]
[2,4,6,8]
[4,6,8,10]
[2,4,6,8,10]
[2,6,10]
```

**Example 2:**

```
Input: nums = [7,7,7,7,7]
Output: 16
Explanation: Any subsequence of this array is arithmetic.
```

**Constraints:**

- `1 <= nums.length <= 1000`
- `-2<sup>31</sup> <= nums[i] <= 2<sup>31</sup> - 1`

- ```python
  class Solution:
      def numberOfArithmeticSlices(self, nums: List[int]) -> int:
          cdDict = {}
          for i in range(len(nums)):
              cdDict[i] = {}
          ans = 0
          for i in range(1,len(cdDict)):
              presentDictionary = cdDict[i]
              for j in range(0,i):
                  cd = nums[i]-nums[j]
                  previousFreq = cdDict[j].get(cd,0)
                  presentfreq  = presentDictionary.get(cd,0)
                  ans+=previousFreq
                  presentDictionary[cd] = presentfreq+previousFreq+1
          # print(cdDict)
          # print(length)
          return ans
  ```
  ```