1027. Longest Arithmetic Subsequence

Given an array `nums` of integers, return the **length** of the longest arithmetic subsequence in `nums`.

Recall that a *subsequence* of an array `nums` is a list `nums[i<sub>1</sub>], nums[i<sub>2</sub>], ..., nums[i<sub>k</sub>]` with `0 <= i<sub>1</sub> < i<sub>2</sub> < ... < i<sub>k</sub> <= nums.length - 1`, and that a sequence `seq` is *arithmetic* if `seq[i+1] - seq[i]` are all the same value (for `0 <= i < seq.length - 1`).

Example 1:

```
Input: nums = [3,6,9,12]
Output: 4
Explanation:
The whole array is an arithmetic sequence with steps of length = 3.
```

Example 2:

```
Input: nums = [9,4,7,2,10]
Output: 3
Explanation:
The longest arithmetic subsequence is [4,7,10].
```

Example 3:

```
Input: nums = [20,1,15,3,10,5,8]
Output: 4
Explanation:
The longest arithmetic subsequence is [20,15,10,5].
```

Constraints:

- `2 <= nums.length <= 1000`
- `0 <= nums[i] <= 500`
- ```
  class Solution:
      def longestArithSeqLength(self, nums: List[int]) -> int:
          cdDict = {}
          for i in range(len(nums)):
              cdDict[i] = {}
          length = 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)
                presentDictionary[cd] = previousFreq+1
                length = max(length,previousFreq+1)
        # print(cdDict)
        # print(length)
        return length+1
```