Contains duplicates

Part 1 using Brute Force

```
from typing import List

class Solution:
    def contains_duplicates(self, nums: List[int]) -> bool:
        for i in range(len(nums)):
            for j in range(i+1, len(nums)):
                if nums[i] == nums[j]:
                    return True
        return False

print(Solution().contains_duplicates(nums=[1,2,3,4,1]))
```

- Step 1:
  self -> solution instance
  nums -> 0 1 2 3 4
          1 2 3 4 1

- Step 2 [for i in range(len(nums)):

i -> 0

- Step 3 (for j in range (i+1, len(nums)):
  j -> 1

- Step 4 (if nums[i] == nums[j]:
  this won't be True, as nums[0]^{i} = 1 and nums[1]^{j} = 2 => 1 == 2 False

- Step 5 (for j in range (i+1, len(nums)):
  we go again in the loop and increase
j -> 2

- Step 6 (if nums[i] == nums[j]:
  again, not True: nums[0] = 1 ✓ / nums[2] = 3 ✓ => False
  1 == 3

- Step 7 (for j in range (i+1, len(nums)):
  we go again in the loop and increase
j -> 3

- Step 8 (if nums[i] == nums[j]:

nums[i] => nums[0] = 1 } => False as nums[i] != [j]
nums[j] => nums[3] = 4 }

- Step 9 (for j in range (i+1, len(nums)):
we go again in the loop and increase j to 4

- Step 10 (if nums[i] == - nums[j]):
i => nums[0] = 1 => nums[i] == nums[j]
j => nums[4] = 1 True

return True ✓