976. Largest Perimeter Triangle

Given an integer array `nums`, return *the largest perimeter of a triangle with a non-zero area, formed from three of these lengths*. If it is impossible to form any triangle of a non-zero area, return `0`.

Example 1:

```
Input: nums = [2,1,2]
Output: 5
```

Example 2:

```
Input: nums = [1,2,1]
Output: 0
```

Example 3:

```
Input: nums = [3,2,3,4]
Output: 10
```

Example 4:

```
Input: nums = [3,6,2,3]
Output: 8
```

Constraints:

- `3 <= nums.length <= 10<sup>4</sup>`
- `1 <= nums[i] <= 10<sup>6</sup>`

```python
class Solution:
    def largestPerimeter(self, nums: List[int]) -> int:
        ans = 0
        n = len(nums)
        nums.sort(reverse=True)
        for i in range(len(nums)-2):
            x = nums[i]
            j = i+1
            k = j+1
            while k<n:
                if x<nums[j]+nums[k]:
                    return x+nums[j]+nums[k]
```

```
                k+=1
        return ans
```

```
class Solution(object):
    def largestPerimeter(self, A):
        A.sort()
        for i in xrange(len(A) - 3, -1, -1):
            if A[i] + A[i+1] > A[i+2]:
                return A[i] + A[i+1] + A[i+2]
        return 0
```