<https://leetcode.com/problems/find-the-smallest-divisor-given-a-threshold/>

Given an array of integers nums and an integer threshold, we will choose a positive integer divisor, divide all the array by it, and sum the division's result. Find the **smallest** divisor such that the result mentioned above is less than or equal to threshold.

Each result of the division is rounded to the nearest integer greater than or equal to that element. (For example: 7/3 = 3 and 10/2 = 5).

The test cases are generated so that there will be an answer.

**Example 1:**

Input: nums = [1,2,5,9], threshold = 6

Output: 5

Explanation: We can get a sum to 17 (1+2+5+9) if the divisor is 1.

If the divisor is 4 we can get a sum of 7 (1+1+2+3) and if the divisor is 5 the sum will be 5 (1+1+1+2).

**Example 2:**

Input: nums = [44,22,33,11,1], threshold = 5

Output: 44

**Constraints:**

1 <= nums.length <= 5 \* 10^4

1 <= nums[i] <= 10^6

nums.length <= threshold <= 10^6

**Attempt 1: 2022-09-16 (10min, problem convert to find the first 'lo' match the requirement, means find the lower boundary of 'lo', use the template directly)**

**Style 1: Auxiliary method not contains 'threshold' parameter**

class Solution {

    public int smallestDivisor(int[] nums, int threshold) {

        int hi = 0;

        for(int num : nums) {

            hi = Math.max(hi, num);

        }

        int lo = 1;

        while(lo <= hi) {

            int mid = lo + (hi - lo) / 2;

            // If sum of division's result larger than threashold means

            // 'mid' (divisor) still smaller than requirement, need to

            // increase 'lo', the problem convert to find the first 'lo'

            // match the requirement, means find the lower boundary of 'lo'

            if(sumOfDivision(nums, mid) > threshold) {

                lo = mid + 1;

            } else {

                hi = mid - 1;

            }

        }

        return lo;

    }

    private int sumOfDivision(int[] nums, int divisor) {

        int count = 0;

        for(int num : nums) {

            count += (num - 1) / divisor + 1;

        }

        return count;

    }

}

Space Complexity: O(1)

Time Complexity: O(nlogn)

**Refer to**

<https://leetcode.com/problems/find-the-smallest-divisor-given-a-threshold/discuss/1001055/JAVA-oror-Binary-Search>

class Solution {

    public int smallestDivisor(int[] nums, int threshold) {

        int left = 1, right = 1000000;

        while(left <= right){

            int mid = left+(right-left)/2;

            long sum = getDivSum(nums, mid);

            if(sum > threshold){

                left = mid + 1;

            } else {

                right = mid - 1;

            }

        }

        return left;

    }

    private long getDivSum(int[] nums, int divisor){

        long sum = 0;

        for(int num: nums){

            sum += (num-1)/divisor+1; //ceil of (num/divisor)

        }

        return sum;

    }

}

**Style 2: Auxiliary method contains 'threshold' parameter**

class Solution {

    public int smallestDivisor(int[] nums, int threshold) {

        int lo = 1;

        int hi = 1;

        for(int num : nums) {

            hi = Math.max(hi, num);

        }

        while(lo <= hi) {

            int mid = lo + (hi - lo) / 2;

            if(getDivSum(nums, threshold, mid)) {

                hi = mid - 1;

            } else {

                lo = mid + 1;

            }

        }

        return lo;

    }

    private boolean getDivSum(int[] nums, int threshold, int smallestDivisor) {

        int count = 0;

        for(int num : nums) {

            count += (num - 1) / smallestDivisor + 1;

            if(count > threshold) {

                return false;

            }

        }

        return true;

    }

}

**Refer to**

[L410.Split Array Largest Sum (Ref.L1011,L704,L1482)](note://WEB4d9d6f26a965381b137615128e2648be)

[L1011.Capacity To Ship Packages Within D Days (Ref.L410,L1482)](note://WEB87ca8f9dd9a14fb4535856a4e0f8f4b1)

[L1482.Minimum Number of Days to Make m Bouquets (Ref.L410,L1011)](note://WEBd5ce394160da45a79e92dbbc806111ec)