SMALLEST DIVISOR WITH A GIVEN THRESHOLD

☆ In this program, we are supposed to find and return the smallest number such that on dividing all the elements by it and adding the ceiling, the value we get is less than threshold.

Brute force solution is to obviously start searching from 1 for each num-ber until we finally find a value Lesser than the threshold. Optimal solution is to achieve this using Binary Search.

Pseudocode :

```
divisorSum(arr, d, t, N) {
    c = 0
    for(i = 0 → N-1) {
        if(arr[i] % d == 0) {
            c += arr[i] / d
        } else {
            c += arr[i] / d + 1
        }
    }
    if(c <= t) {
        return true
    } else {
        return false
    }
}
```

```
smallestDivisor(arr, N, t) {
    max = 0
    for(i = 1 -> N - 1) {
        if (arr[i] > arr[max]) {
            max = i
        }
    }
    low = 1
    ans = -1
    max = arr[max]
    while(low <= max) {
        mid = (low + max) / 2
        if(divisorSum(arr, mid, t, N)) {
            ans = mid
            max = mid - 1
        } else {
            low = mid + 1
        }
    }
    return ans
}
```