

长度无限长的公路上，小美雇佣了 n 位工人来种树，每个点最多种一棵树。从左向右数，工人所站的位置为 a_1, a_2, \dots, a_n 。已知每位工人都会将自己所在位置的右侧段长度的区间种满树，且每位工人的种树区间长度相同。现在小美希望公路上至少有 k 棵树，为了节约成本，他希望每位工人种树的区间长度尽可能请你帮他求出，工人们的种树区间至少多长，才能使得公路被种上至少 k 棵树。

输入描述

第一行输入两个正整数 $n, k (1 < n, k < 2 \times 10^5)$ ，分别表示工人的数量，以及小美要求树的最少数量。第二行输入几个正整数 $a_1, a_2, \dots, a_n (1 \leq a_i \leq 2 \times 10^5)$ ，表示每名工人的位置，

输出描述

在一行上输出一个整数，代表工人们最短的种树区间长度。

输入

3 6

1 2 5

输出

3

说明

每位工人种树的区间长度至少为 3.

这样以来:

第一名工人种:1,2,3 点的树。

第二名工人种:2,3,4 点的树,

第三名工人种:5,6,7 点的树。

由于每个位置最多种一棵树，因此共有:1,2,3,4,5,6,7 这些点有树，满足至少 $k = 6$ 棵树。

可以证明，不存在比 3 更小的答案

思路:

二分+区间合并

代码答案:

```

import sys
input=lambda:sys.stdin.readline().strip()
def solve(n, k, positions):
    positions.sort()
    low, high = 1, k

    def check(mid):
        a = []
        b = []
        ans = 0
        for c in positions:
            a.append([c, c + mid - 1])
        for l, r in a:
            if not b or l > b[-1][1]:
                b.append([l, r])
            else:
                b[-1][1] = max(b[-1][1], r)

        for l, r in b:
            ans += r - l + 1
        return ans >= k
    while low < high:
        mid = (low + high) // 2
        if check(mid):
            high = mid
        else:
            low = mid + 1

    return low
n, k = map(int, input().split())
positions = list(map(int, input().split()))
print(solve(n, k, positions))

```