Problem 1891: Cutting Ribbons

Problem Information

Difficulty: Medium
Acceptance Rate: 52.95%
Paid Only: Yes
Tags: Array, Binary Search

Problem Description

You are given an integer array `ribbons`, where `ribbons[i]` represents the length of the `ith` ribbon, and an integer `k`. You may cut any of the ribbons into any number of segments of **positive integer** lengths, or perform no cuts at all.

* For example, if you have a ribbon of length `4`, you can: * Keep the ribbon of length `4`, * Cut it into one ribbon of length `3` and one ribbon of length `1`, * Cut it into two ribbons of length `2`, * Cut it into one ribbon of length `2` and two ribbons of length `1`, or * Cut it into four ribbons of length `1`.

Your task is to determine the **maximum** length of ribbon, `x`, that allows you to cut _at least_ `k` ribbons, each of length `x`. You can discard any leftover ribbon from the cuts. If it is **impossible** to cut `k` ribbons of the same length, return 0.

Example 1:

Input: ribbons = [9,7,5], k = 3 **Output:** 5 **Explanation:** - Cut the first ribbon to two ribbons, one of length 5 and one of length 4. - Cut the second ribbon to two ribbons, one of length 5 and one of length 2. - Keep the third ribbon as it is. Now you have 3 ribbons of length 5.

Example 2:

Input: ribbons = [7,5,9], k = 4 **Output:** 4 **Explanation:** - Cut the first ribbon to two ribbons, one of length 4 and one of length 3. - Cut the second ribbon to two ribbons, one of length 4 and one of length 1. - Cut the third ribbon to three ribbons, two of length 4 and one of length 1. Now you have 4 ribbons of length 4.

Example 3:

Input: ribbons = [5,7,9], k = 22 **Output:** 0 **Explanation:** You cannot obtain k ribbons of the same positive integer length.

Constraints:

* `1 <= ribbons.length <= 105` * `1 <= ribbons[i] <= 105` * `1 <= k <= 109`

Code Snippets

C++:

```
class Solution {
public:
int maxLength(vector<int>& ribbons, int k) {

}
};
```

Java:

```
class Solution {
public int maxLength(int[] ribbons, int k) {

}
}
```

Python3:

```
class Solution:
def maxLength(self, ribbons: List[int], k: int) -> int:
```