Problem 1566: Detect Pattern of Length M Repeated K or More Times

Problem Information

Difficulty: Easy
Acceptance Rate: 43.65%
Paid Only: No
Tags: Array, Enumeration

Problem Description

Given an array of positive integers `arr`, find a pattern of length `m` that is repeated `k` or more times.

A **pattern** is a subarray (consecutive sub-sequence) that consists of one or more values, repeated multiple times **consecutively** without overlapping. A pattern is defined by its length and the number of repetitions.

Return `true` _if there exists a pattern of length_ `m` _that is repeated_ `k` _or more times, otherwise return_ `false`.

Example 1:

Input: arr = [1,2,4,4,4,4], m = 1, k = 3 **Output:** true **Explanation:** The pattern **(4)** of length 1 is repeated 4 consecutive times. Notice that pattern can be repeated k or more times but not less.

Example 2:

Input: arr = [1,2,1,2,1,1,1,3], m = 2, k = 2 **Output:** true **Explanation:** The pattern **(1,2)** of length 2 is repeated 2 consecutive times. Another valid pattern **(2,1) is** also repeated 2 times.

Example 3:

Input: arr = [1,2,1,2,1,3], m = 2, k = 3 **Output:** false **Explanation:** The pattern (1,2) is of length 2 but is repeated only 2 times. There is no pattern of length 2 that is repeated 3 or more times.

Constraints:

* `2 <= arr.length <= 100` * `1 <= arr[i] <= 100` * `1 <= m <= 100` * `2 <= k <= 100`

Code Snippets

C++:

```
class Solution {
public:
bool containsPattern(vector<int>& arr, int m, int k) {

}
};
```

Java:

```
class Solution {
public boolean containsPattern(int[] arr, int m, int k) {

}
}
```

Python3:

```
class Solution:
def containsPattern(self, arr: List[int], m: int, k: int) -> bool:
```