5497. Find Latest Group of Size M

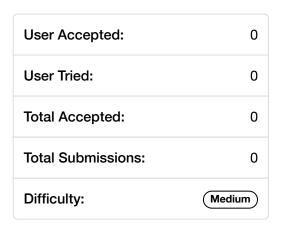
My Submissions (/contest/weekly-contest-203/problems/find-latest-group-of-size-m/submissions/)

Back to Contest (/contest/weekly-contest-203/)

Given an array arr that represents a permutation of numbers from 1 to n. You have a binary string of size n that initially has all its bits set to zero.

At each step i (assuming both the binary string and arr are 1-indexed) from 1 to n, the bit at position arr[i] is set to 1. You are given an integer m and you need to find the latest step at which there exists a group of ones of length m. A group of ones is a contiguous substring of 1s such that it cannot be extended in either direction.

Return the latest step at which there exists a group of ones of length **exactly** m . If no such group exists, return -1.



Example 1:

```
Input: arr = [3,5,1,2,4], m = 1
Output: 4
Explanation:
Step 1: "00100", groups: ["1"]
Step 2: "00101", groups: ["1", "1"]
Step 3: "10101", groups: ["1", "1"]
Step 4: "11101", groups: ["111", "1"]
Step 5: "11111", groups: ["11111"]
The latest step at which there exists a group of size 1 is step 4.
```

Example 2:

```
Input: arr = [3,1,5,4,2], m = 2
Output: -1
Explanation:
Step 1: "00100", groups: ["1"]
Step 2: "10100", groups: ["1", "1"]
Step 3: "10101", groups: ["1", "1"]
Step 4: "10111", groups: ["1", "111"]
Step 5: "11111", groups: ["11111"]
No group of size 2 exists during any step.
```

Example 3:

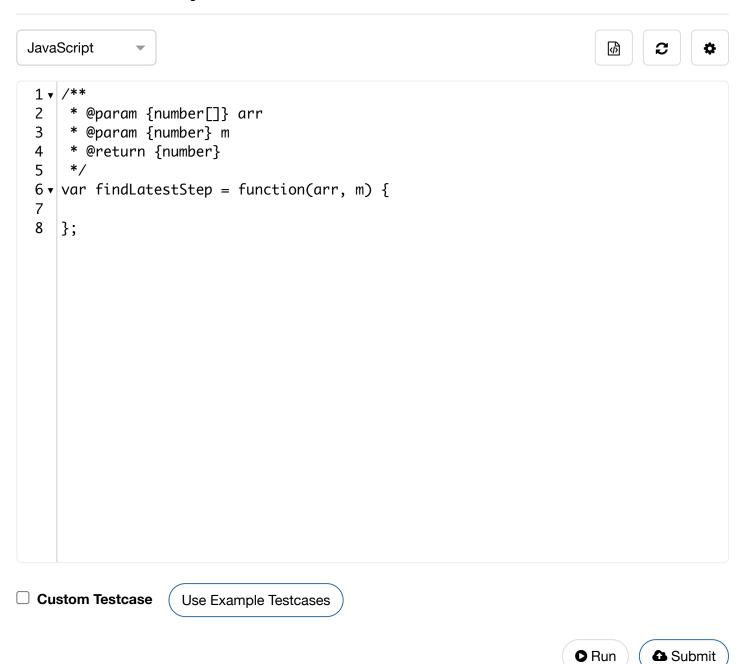
```
Input: arr = [1], m = 1
Output: 1
```

Example 4:

```
Input: arr = [2,1], m = 2
Output: 2
```

Constraints:

- n == arr.length
- 1 <= n <= 10⁵
- 1 <= arr[i] <= n
- All integers in arr are distinct.
- 1 <= m <= arr.length



Help Center (/support/) | Terms (/terms/) | Privacy Policy (/privacy/)

States (/region/)