Problem 2612: Minimum Reverse Operations

Problem Information

Difficulty: Hard
Acceptance Rate: 16.13%
Paid Only: No
Tags: Array, Hash Table, Breadth-First Search, Union Find, Ordered Set

Problem Description

You are given an integer `n` and an integer `p` representing an array `arr` of length `n` where all elements are set to 0's, except position `p` which is set to 1. You are also given an integer array `banned` containing restricted positions. Perform the following operation on `arr`:

* Reverse a **subarray** with size `k` if the single 1 is not set to a position in `banned`.

Return an integer array `answer` with `n` results where the `ith` result is __ the **minimum** number of operations needed to bring the single 1 to position `i` in `arr`, or -1 if it is impossible.

Example 1:

Input: n = 4, p = 0, banned = [1,2], k = 4

Output: [0,-1,-1,1]

Explanation:

* Initially 1 is placed at position 0 so the number of operations we need for position 0 is 0. * We can never place 1 on the banned positions, so the answer for positions 1 and 2 is -1. * Perform the operation of size 4 to reverse the whole array. * After a single operation 1 is at position 3 so the answer for position 3 is 1.

Example 2:

Input: n = 5, p = 0, banned = [2,4], k = 3

Output: [0,-1,-1,-1,-1]

Explanation:

* Initially 1 is placed at position 0 so the number of operations we need for position 0 is 0. * We cannot perform the operation on the subarray positions `[0, 2]` because position 2 is in banned. * Because 1 cannot be set at position 2, it is impossible to set 1 at other positions in more operations.

Example 3:

Input: n = 4, p = 2, banned = [0,1,3], k = 1

Output: [-1,-1,0,-1]

Explanation:

Perform operations of size 1 and 1 never changes its position.

Constraints:

* `1 <= n <= 105` * `0 <= p <= n - 1` * `0 <= banned.length <= n - 1` * `0 <= banned[i] <= n - 1` * `1 <= k <= n ` * `banned[i] != p` * all values in `banned` are **unique**

Code Snippets

C++:

```
class Solution {
public:
vector<int> minReverseOperations(int n, int p, vector<int>& banned, int k) {

}
};
```

Java:

```
class Solution {
public int[] minReverseOperations(int n, int p, int[] banned, int k) {
```

```
}
}
```

Python3:

```
class Solution:
def minReverseOperations(self, n: int, p: int, banned: List[int], k: int) ->
List[int]:
```