Problem 3037: Find Pattern in Infinite Stream II

Problem Information

Difficulty: Hard
Acceptance Rate: 66.91%
Paid Only: Yes
Tags: Array, Sliding Window, Rolling Hash, String Matching, Hash Function

Problem Description

You are given a binary array `pattern` and an object `stream` of class `InfiniteStream` representing a **0-indexed** infinite stream of bits.

The class `InfiniteStream` contains the following function:

* `int next()`: Reads a **single** bit (which is either `0` or `1`) from the stream and returns it.

Return _the**first starting** index where the pattern matches the bits read from the stream_. For example, if the pattern is `[1, 0]`, the first match is the highlighted part in the stream `[0, **_1, 0_** , 1, ...]`.

Example 1:

Input: stream = [1,1,1,0,1,1,1,...], pattern = [0,1] **Output:** 3 **Explanation:** The first occurrence of the pattern [0,1] is highlighted in the stream [1,1,1,**_0,1_** ,...], which starts at index 3.

Example 2:

Input: stream = [0,0,0,0,...], pattern = [0] **Output:** 0 **Explanation:** The first occurrence of the pattern [0] is highlighted in the stream [**_0_** ,...], which starts at index 0.

Example 3:

Input: stream = [1,0,1,1,0,1,1,0,1,...], pattern = [1,1,0,1] **Output:** 2 **Explanation:** The first occurrence of the pattern [1,1,0,1] is highlighted in the stream [1,0,**_1,1,0,1_** ,...],

which starts at index 2.

Constraints:

* `1 <= pattern.length <= 104` * `pattern` consists only of `0` and `1`. * `stream` consists only of `0` and `1`. * The input is generated such that the pattern's start index exists in the first `105` bits of the stream.

Code Snippets

C++:

```
/**
* Definition for an infinite stream.
* class InfiniteStream {
* public:
* InfiniteStream(vector<int> bits);
* int next();
* };
*/
class Solution {
public:
int findPattern(InfiniteStream* stream, vector<int>& pattern) {

}
};
```

Java:

```
/**
* Definition for an infinite stream.
* class InfiniteStream {
* public InfiniteStream(int[] bits);
* public int next();
* }
*/
class Solution {
public int findPattern(InfiniteStream infiniteStream, int[] pattern) {

}
}
```

Python3:

```
# Definition for an infinite stream.
# class InfiniteStream:
# def next(self) -> int:
# pass
class Solution:
def findPattern(self, stream: Optional['InfiniteStream'], pattern: List[int])
-> int:
```