Problem 281: Zigzag Iterator

Problem Information

Difficulty: Medium
Acceptance Rate: 66.26%
Paid Only: Yes
Tags: Array, Design, Queue, Iterator

Problem Description

Given two vectors of integers `v1` and `v2`, implement an iterator to return their elements alternately.

Implement the `ZigzagIterator` class:

* `ZigzagIterator(List<int> v1, List<int> v2)` initializes the object with the two vectors `v1` and `v2`. * `boolean hasNext()` returns `true` if the iterator still has elements, and `false` otherwise. * `int next()` returns the current element of the iterator and moves the iterator to the next element.

Example 1:

Input: v1 = [1,2], v2 = [3,4,5,6] **Output:** [1,3,2,4,5,6] **Explanation:** By calling next repeatedly until hasNext returns false, the order of elements returned by next should be: [1,3,2,4,5,6].

Example 2:

Input: v1 = [1], v2 = [] **Output:** [1]

Example 3:

Input: v1 = [], v2 = [1] **Output:** [1]

Constraints:

* `0 <= v1.length, v2.length <= 1000` * `1 <= v1.length + v2.length <= 2000` * `-231 <= v1[i], v2[i] <= 231 - 1`

Follow up: What if you are given `k` vectors? How well can your code be extended to such cases?

Clarification for the follow-up question:

The "Zigzag" order is not clearly defined and is ambiguous for `k > 2` cases. If "Zigzag" does not look right to you, replace "Zigzag" with "Cyclic".

Follow-up Example:

Input: v1 = [1,2,3], v2 = [4,5,6,7], v3 = [8,9] **Output:** [1,4,8,2,5,9,3,6,7]

Code Snippets

C++:

```
class ZigzagIterator {
public:
ZigzagIterator(vector<int>& v1, vector<int>& v2) {

}

int next() {

}

bool hasNext() {

}
};

/**
* Your ZigzagIterator object will be instantiated and called as such:
* ZigzagIterator i(v1, v2);
* while (i.hasNext()) cout << i.next();
*/
```

Java:

```
public class ZigzagIterator {

public ZigzagIterator(List<Integer> v1, List<Integer> v2) {

}

public int next() {

}

public boolean hasNext() {

}
}

/**
* Your ZigzagIterator object will be instantiated and called as such:
* ZigzagIterator i = new ZigzagIterator(v1, v2);
* while (i.hasNext()) v[f()] = i.next();
*/
```

Python3:

```
class ZigzagIterator:
def __init__(self, v1: List[int], v2: List[int]):


def next(self) -> int:


def hasNext(self) -> bool:


# Your ZigzagIterator object will be instantiated and called as such:
# i, v = ZigzagIterator(v1, v2), []
# while i.hasNext(): v.append(i.next())
```