

281. Zigzag Iterator

Medium 347 20 Add to List Share

Given two 1d vectors, implement an iterator to return their elements alternately.

Example:

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].

Follow up:

What if you are given k 1d 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". For example:

Input:
 [1,2,3]
 [4,5,6,7]
 [8,9]
 Output: [1,4,8,2,5,9,3,6,7].

Accepted 61,916 Submissions 107,056

Seen this question in a real interview before?

Yes

No

Contributor



Companies i



Related Topics



Similar Questions



i {} > < > <

```

1 class ZigzagIterator(object):
2
3     def __init__(self, v1, v2):
4         """
5         Initialize your data
6         structure here.
7         :type v1: List[int]
8         :type v2: List[int]
9         """
10
11     def next(self):
12         """
13         :rtype: int
14         """
15
16     def hasNext(self):
17         """
18         :rtype: bool
19         """
20
21
22
23 # Your ZigzagIterator object will
24 # be instantiated and called as
25 # such:
26 # i, v = ZigzagIterator(v1, v2),
27 # []
28 # while i.hasNext():
29 #     v.append(i.next())

```

Console

Contribute i