Zigzag Iterator

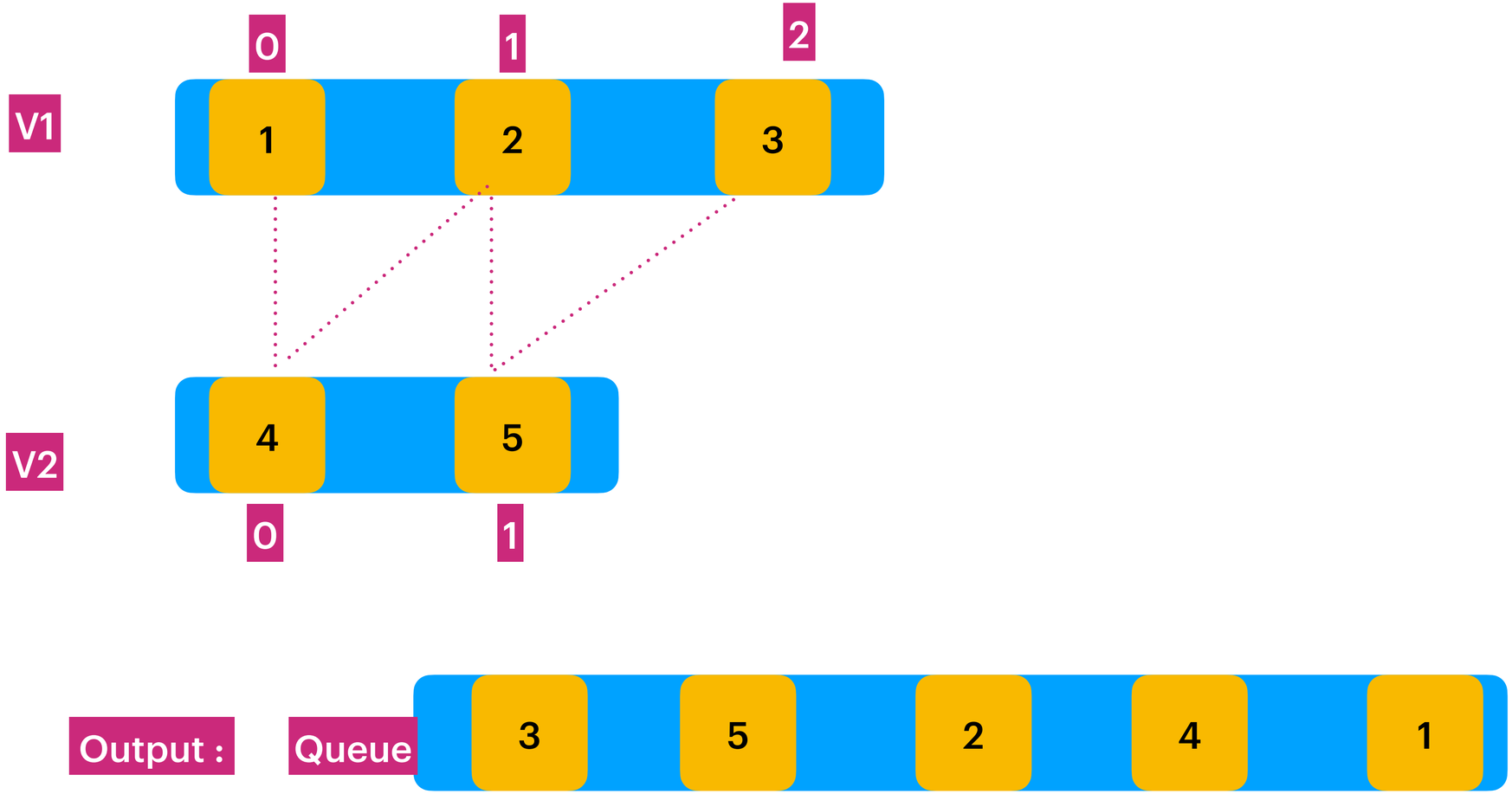
Given two vectors of integers v1 and v2, implement an iterator to return their elements alternately.

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.

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].

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


0
1
2
V1
1
2
3
4
5
V2
0
1
Output :
Queue
3
5
2
4
1

Time Needed to Buy Tickets

There are n people in a line queuing to buy tickets, where the 0th person is at the front of the line and the (n - 1)th person is at the back of the line.

You are given a 0-indexed integer array tickets of length n where the number of tickets that the ith person would like to buy is tickets[i].

Each person takes exactly 1 second to buy a ticket. A person can only buy 1 ticket at a time and has to go back to the end of the line (which happens instantaneously) in order to buy more tickets. If a person does not have any tickets left to buy, the person will leave the line.

Return the time taken for the person at position k (0-indexed) to finish buying tickets.

Input: tickets = [2,3,2], k = 2
Output: 6
Explanation:
- In the first pass, everyone in the line buys a ticket and the line becomes [1, 2, 1].
- In the second pass, everyone in the line buys a ticket and the line becomes [0, 1, 0].
The person at position 2 has successfully bought 2 tickets and it took 3 + 3 = 6 seconds.

Input: tickets = [2,3,2], k = 2
Output: 6
Explanation:
- In the first pass, everyone in the line buys a ticket and the line becomes [1, 2, 1].
- In the second pass, everyone in the line buys a ticket and the line becomes [0, 1, 0].
The person at position 2 has successfully bought 2 tickets and it took 3 + 3 = 6 seconds.

tickets = [2[0],3[1],2[2]] k = 2

[2[0],3[1],2[2]] 0sec 2[2] -> 3[1] -> 2[0] [Front] : counter = 0

[3[1],2[2],1[0]] 1Sec 1[0] -> 2[2] -> 3[1] [Front] : counter = 1

[2[2],1[0],2[1]] 2Sec 2[1] -> 1[0] -> 2[2] [Front] : counter = 2

[1[0],2[1],1[2]] 3Sec 1[2] -> 2[1] -> 1[0] [Front] : counter = 3

[2[1],1[2],0[0](X)] 4Sec 0[0](X) don't add to deque
as the value zero

[2[1],1[2]] 4Sec 1[2] -> 2[1] [Front] : counter = 4

[1[2],1[1]] 5Sec 1[1] -> 1[2] [Front] : counter = 5

[1[1],0[2] (X)] 6Sec 0[2] (X) as k == 2 & value = 0 return counter : 6
1[1] [Front] : counter = 6

tickets = [2[0],3[1],2[2]] k = 1

[2[0],3[1],2[2]] 0sec

[3[1],2[2],1[0]] 1Sec

[2[2],1[0],2[1]] 2Sec

[1[0],2[1],1[2]] 3Sec

[2[1],1[2],0[0](X)] 4Sec

[2[1],1[2]] 4Sec

[1[2],1[1]] 5Sec

[1[1],0[2] (X)] 6Sec

[1[1]] 6Sec

[0[1] X] 7Sec

[] 7Sec