Our Solution(s)

Run Code

Solution 1

```go
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

package main

// O(n) time | O(1) space - where n is the length of the input array
func HasSingleCycle(array []int) bool {
  numElementsVisited := 0
  currentIdx := 0
  for numElementsVisited < len(array) {
    if numElementsVisited > 0 && currentIdx == 0 {
      return false
    }
    numElementsVisited++
    currentIdx = getNextIdx(currentIdx, array)
  }
  return currentIdx == 0
}

func getNextIdx(currentIdx int, array []int) int {
  jump := array[currentIdx]
  nextIdx := (currentIdx + jump) % len(array)
  if nextIdx >= 0 {
    return nextIdx
  }
  return nextIdx + len(array)
}
```

Your Solutions

Run Code

Solution 1   Solution 2   Solution 3

```go
package main

func HasSingleCycle(array []int) bool {
  // Write your code here.
  return false
}
```

Custom Output   Raw Output   Submit Code

Run or submit code when you're ready.