Prompt | Scratchpad | Our Solution(s) | Video Explanation | Run Code

Solution 1

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

public class Program {
  // O(n) time | O(1) space - where n is the length of the input array
  public static bool HasSingleCycle(int[] array) {
    int numElementsVisited = 0;
    int currentIdx = 0;
    while (numElementsVisited < array.Length) {
      if (numElementsVisited > 0 && currentIdx == 0) return false;
      numElementsVisited++;
      currentIdx = getNextIdx(currentIdx, array);
    }
    return currentIdx == 0;
  }

  public static int getNextIdx(int currentIdx, int[] array) {
    int jump = array[currentIdx];
    int nextIdx = (currentIdx + jump) % array.Length;
    return nextIdx >= 0 ? nextIdx : nextIdx + array.Length;
  }
}

```

Your Solutions | Run Code

Solution 1 | Solution 2 | Solution 3

```
public class Program {
  public static bool HasSingleCycle(int[] array) {
    // Write your code here.
    return false;
  }
}

```

Custom Output | Raw Output | Submit Code

Run or submit code when you're ready.