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 LongestPeak(array []int) int {
  longestPeakLength := 0
  i := 1
  for i < len(array)-1 {
    isPeak := array[i-1] < array[i] && array[i] > array[i+1]
    if !isPeak {
      i += 1
      continue
    }

    leftIdx := i - 2
    for leftIdx >= 0 && array[leftIdx] < array[leftIdx+1] {
      leftIdx -= 1
    }

    rightIdx := i + 2
    for rightIdx < len(array) && array[rightIdx] < array[rightIdx-1] {
      rightIdx += 1
    }
    currentPeakLength := rightIdx - leftIdx - 1
    if currentPeakLength > longestPeakLength {
      longestPeakLength = currentPeakLength
    }
    i = rightIdx
  }
  return longestPeakLength
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```go
package main

func LongestPeak(array []int) int {
  // Write your code here.
  return -1
}

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.