Our Solution(s) Run Code

Solution 1

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

package main

import "math"

// O(n^2) time | O(n) space
func MaxSumIncreasingSubsequence(array []int) []interface{} {
  sequences := make([]int, len(array))
  sums := make([]int, len(array))
  for i := range sequences {
    sequences[i] = math.MinInt32
    sums[i] = array[i]
  }
  maxSumIndex := 0
  for i, currentNum := range array {
    for j := 0; j < i; j++ {
      otherNum := array[j]
      if otherNum < currentNum && sums[j]+currentNum >= sums[i] {
        sums[i] = sums[j] + currentNum
        sequences[i] = j
      }
    }
    if sums[i] > sums[maxSumIndex] {
      maxSumIndex = i
    }
  }

  maxSum := sums[maxSumIndex]
  sequence := buildSequence(array, sequences, maxSumIndex)
  return []interface{}{maxSum, sequence}
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```go
package main

func MaxSumIncreasingSubsequence(array []int) []interface{} {
  // Write your code here.
  return []interface{}{
    10,             // Example max sum
    []int{2, 3, 5}, // Example max sequence
  }
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.