

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 import "math"
6
7 // O(n^2) time | O(n) space
8 func MaxSumIncreasingSubsequence(array []int) []interface{} {
9     sequences := make([]int, len(array))
10    sums := make([]int, len(array))
11    for i := range sequences {
12        sequences[i] = math.MinInt32
13        sums[i] = array[i]
14    }
15    maxSumIndex := 0
16    for i, currentNum := range array {
17        for j := 0; j < i; j++ {
18            otherNum := array[j]
19            if otherNum < currentNum && sums[j]+currentNum >= sums[i] {
20                sums[i] = sums[j] + currentNum
21                sequences[i] = j
22            }
23        }
24        if sums[i] > sums[maxSumIndex] {
25            maxSumIndex = i
26        }
27    }
28
29    maxSum := sums[maxSumIndex]
30    sequence := buildSequence(array, sequences, maxSumIndex)
31    return []interface{}{maxSum, sequence}
32 }
33
```

Solution 1 Solution 2 Solution 3

```
1 package main
2
3 func MaxSumIncreasingSubsequence(array []int) []interface{} {
4     // Write your code here.
5     return []interface{}{
6         10, // Example max sum
7         []int{2, 3, 5}, // Example max sequence
8     }
9 }
10
```

Our Tests

Custom Output

Submit Code

```
11 Run in TestRunner TestRunner TestRunner 2
12 Update Report, RelativeAccuracyReport(200, 50, 1000000)
13 2
14
15 Run in TestRunner TestRunner TestRunner 2
16 Update Report, RelativeAccuracyReport(200, 50, 1000000)
17 2
18
19 Run in TestRunner TestRunner TestRunner 2
20 Update Report, RelativeAccuracyReport(200, 5, 50, 1000000)
21 2
22
23 Run in TestRunner TestRunner TestRunner 2
24 Update Report, RelativeAccuracyReport(200, 5, 5, 5, 5, 50, 1000000)
```

Run or submit code when you're ready.