

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n^2) time | O(n) space
5     func maximumSumIncreasingSubsequence(array: [Int]) -> (Int, [Int]) {
6         var maxSumIndex = 0
7         var sums = array.map { $0 }
8         var previousIndices: [Int?] = Array(repeating: nil, count: array.count)
9
10        for i in 0 ..< array.count {
11            let currentNumber = array[i]
12            for j in 0 ..< i {
13                let previousNumber = array[j]
14                if previousNumber < currentNumber, sums[j] + currentNumber > sums[i] {
15                    sums[i] = sums[j] + currentNumber
16                    previousIndices[i] = j
17                }
18            }
19
20            if sums[i] > sums[maxSumIndex] {
21                maxSumIndex = i
22            }
23        }
24
25        return (sums[maxSumIndex], buildSequence(array, maxSumIndex, previousIndices))
26    }
27
28    func buildSequence(_ array: [Int], _ maxSumIndex: Int, _ previousIndices: [Int?]) -> [Int] {
29        var sequence = [Int]()
30        var currentIndex: Int? = maxSumIndex
31
32        while currentIndex != nil {
33            sequence.insert(array[currentIndex!], at: 0)
34            currentIndex = previousIndices[currentIndex!]
```

Solution 1

Solution 2

Solution 3

```
1 class Program {
2     func maximumSumIncreasingSubsequence(array: [Int]) -> (Int, [Int]) {
3         // Write your code here.
4         return (-1, [])
5     }
6 }
7
```

```
1 class Program {
2     func maximumSumIncreasingSubsequence(array: [Int]) -> (Int, [Int]) {
3         // Write your code here.
4         return (-1, [])
5     }
6 }
7
```

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n^2) time | O(n) space
5     func maximumSumIncreasingSubsequence(array: [Int]) -> (Int, [Int]) {
6         var maxSumIndex = 0
7         var sums = array.map { $0 }
8         var previousIndices: [Int?] = Array(repeating: nil, count: array.count)
9
10        for i in 0 ..< array.count {
11            let currentNumber = array[i]
12            for j in 0 ..< i {
13                let previousNumber = array[j]
14                if previousNumber < currentNumber, sums[j] + currentNumber > sums[i] {
15                    sums[i] = sums[j] + currentNumber
16                    previousIndices[i] = j
17                }
18            }
19
20            if sums[i] > sums[maxSumIndex] {
21                maxSumIndex = i
22            }
23        }
24
25        return (sums[maxSumIndex], buildSequence(array, maxSumIndex, previousIndices))
26    }
27
28    func buildSequence(_ array: [Int], _ maxSumIndex: Int, _ previousIndices: [Int?]) -> [Int] {
29        var sequence = [Int]()
30        var currentIndex: Int? = maxSumIndex
31
32        while currentIndex != nil {
33            sequence.insert(array[currentIndex!], at: 0)
34            currentIndex = previousIndices[currentIndex!]
```

```
17 # my_submission.py
18
19 """Test Case 1"""
20 # my_submission.py
21
22 """Test Case 2"""
23 # my_submission.py
24
25 """Test Case 3"""
26 # my_submission.py
27
28 """Test Case 4"""
29 # my_submission.py
30
31 """Test Case 5"""
32 # my_submission.py
```

Run or submit code when you're ready.