

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 import java.util.*;
4
5 class Program {
6     // O(n^2) time | O(n) space
7     public static List<Integer> longestIncreasingSubsequence(int[] array) {
8         int[] sequences = new int[array.length];
9         Arrays.fill(sequences, Integer.MIN_VALUE);
10        int[] lengths = new int[array.length];
11        Arrays.fill(lengths, 1);
12        int maxLengthIdx = 0;
13        for (int i = 0; i < array.length; i++) {
14            int currentNum = array[i];
15            for (int j = 0; j < i; j++) {
16                int otherNum = array[j];
17                if (otherNum < currentNum && lengths[j] + 1 >= lengths[i]) {
18                    lengths[i] = lengths[j] + 1;
19                    sequences[i] = j;
20                }
21            }
22            if (lengths[i] >= lengths[maxLengthIdx]) {
23                maxLengthIdx = i;
24            }
25        }
26        return buildSequence(array, sequences, maxLengthIdx);
27    }
28
29    public static List<Integer> buildSequence(int[] array, int[] sequences, int currentIndex) {
30        List<Integer> sequence = new ArrayList<Integer>();
31        while (currentIndex != Integer.MIN_VALUE) {
32            sequence.add(0, array[currentIndex]);
33            currentIndex = sequences[currentIndex];
34        }
35        return sequence;
36    }
37 }
```

Solution 1

Solution 2

Solution 3

```
1 import java.util.*;
2
3 class Program {
4     public static List<Integer> longestIncreasingSubsequence(int[] array) {
5         // Write your code here.
6         return null;
7     }
8 }
9
```

Our Tests

Custom Output

Submit Code

```
1 import java.util.*;
2
3 class Program {
4     // Write your code here.
5     return null;
6 }
```

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 import java.util.*;
4
5 class Program {
6     // O(n^2) time | O(n) space
7     public static List<Integer> longestIncreasingSubsequence(int[] array) {
8         int[] sequences = new int[array.length];
9         Arrays.fill(sequences, Integer.MIN_VALUE);
10        int[] lengths = new int[array.length];
11        Arrays.fill(lengths, 1);
12        int maxLengthIdx = 0;
13        for (int i = 0; i < array.length; i++) {
14            int currentNum = array[i];
15            for (int j = 0; j < i; j++) {
16                int otherNum = array[j];
17                if (otherNum < currentNum && lengths[j] + 1 >= lengths[i]) {
18                    lengths[i] = lengths[j] + 1;
19                    sequences[i] = j;
20                }
21            }
22            if (lengths[i] >= lengths[maxLengthIdx]) {
23                maxLengthIdx = i;
24            }
25        }
26        return buildSequence(array, sequences, maxLengthIdx);
27    }
28
29    public static List<Integer> buildSequence(int[] array, int[] sequences, int currentIndex) {
30        List<Integer> sequence = new ArrayList<Integer>();
31        while (currentIndex != Integer.MIN_VALUE) {
32            sequence.add(0, array[currentIndex]);
33            currentIndex = sequences[currentIndex];
34        }
35        return sequence;
36    }
37 }
```

```
14
15
16 @test
17 test test_hasall() {
18     test_expect_fail = 1 1 1 1 1
19     test_expect_fail test_hasall test_hasall test_hasall test_hasall test_hasall
20 }
21
22
23 @test
24 test
25 test test_hasall() {
26     test_expect_fail = 1 1 1 1 1
27     test_expect_fail
28     test_hasall test_hasall test_hasall test_hasall test_hasall test_hasall test_hasall test_hasall test_hasall test_hasall
29 }
30
```

Run or submit code when you're ready.