

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <climits>
5 using namespace std;
6
7 vector<int> buildSequence(vector<int> array, vector<int> sequences,
8                           int currentIdx);
9
10 // O(n^2) time | O(n) space
11 vector<int> longestIncreasingSubsequence(vector<int> array) {
12     vector<int> sequences(array.size(), INT_MIN);
13     vector<int> lengths(array.size(), 1);
14     int maxLengthIdx = 0;
15     for (int i = 0; i < array.size(); i++) {
16         int currentNum = array[i];
17         for (int j = 0; j < i; j++) {
18             int otherNum = array[j];
19             if (otherNum < currentNum && lengths[j] + 1 >= lengths[i]) {
20                 lengths[i] = lengths[j] + 1;
21                 sequences[i] = j;
22             }
23         }
24         if (lengths[i] >= lengths[maxLengthIdx]) {
25             maxLengthIdx = i;
26         }
27     }
28     return buildSequence(array, sequences, maxLengthIdx);
29 }
30
31 vector<int> buildSequence(vector<int> array, vector<int> sequences,
32                           int currentIdx) {
33     vector<int> sequence;
```

Solution 1Solution 2Solution 3

```
1 #include <vector>
2 using namespace std;
3
4 vector<int> longestIncreasingSubsequence(vector<int> array) {
5     // Write your code here.
6     return {};
7 }
8
```

Our Tests

Custom Output

Submit Code

```
10 vector<int> vector1{0, 1, 2, 3, 4, 5};
11 vector<int> vector2{0, 1, 2, 3, 4, 5, 6, 7};
12 vector<int> vector3{0, 1, 2, 3, 4, 5};
13
14 // Copy Constructor - public: Vector() {
15 //     ...
16 //     ...
17 // }
18
19 // Copy Constructor - public: Vector(const Vector& v) {
20 //     ...
21 //     ...
22 // }
23
24 // Copy Constructor - public: Vector(const Vector& v, int n) {
25 //     ...
26 //     ...
27 // }
```

Run or submit code when you're ready.