

AlgoExpert

Quad Layout

C++

12px

Sublime

Monokai

00:00:00

PromptScratchpadOur Solution(s)Video Explanation

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 int binarySearch(int startIdx, int endIdx, vector<int> indices,
8 vector<int> array, int num);
9 vector<int> buildSequence(vector<int> array, vector<int> sequences,
10 int currentIdx);
11
12 // O(nlogn) time | O(n) space
13 vector<int> longestIncreasingSubsequence(vector<int> array) {
14 vector<int> sequences(array.size(), 0);
15 vector<int> indices(array.size() + 1, INT_MIN);
16 int length = 0;
17 for (int i = 0; i < array.size(); i++) {
18 int num = array[i];
19 int newLength = binarySearch(1, length, indices, array, num);
20 sequences[i] = indices[newLength - 1];
21 indices[newLength] = i;
22 length = max(length, newLength);
23 }
24 return buildSequence(array, sequences, indices[length]);
25 }
26
27 int binarySearch(int startIdx, int endIdx, vector<int> indices,
28 vector<int> array, int num) {
29 if (startIdx > endIdx) {
30 return startIdx;
31 }
32 int middleIdx = (startIdx + endIdx) / 2;
33 if (array[indices[middleIdx]] < num) {
34 startIdx = middleIdx + 1;
35 } else {
36 endIdx = middleIdx - 1;
37 }
38 return binarySearch(startIdx, endIdx, indices, array, num);
39 }
40
41 vector<int> buildSequence(vector<int> array, vector<int> sequences,
42 int currentIdx) {
43 vector<int> sequence;
44 while (currentIdx != INT_MIN) {
45 sequence.insert(sequence.begin(), array[currentIdx]);
46 currentIdx = sequences[currentIdx];
47 }
48 return sequence;
49 }
50

