AlgoExpert Quad Layout C++ 12px Sublime Monok

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1 Solution 2 Solution 3

```cpp
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

using namespace std;

class SpecialBST {
public:
  int value;
  int idx;
  int numSmallerAtInsertTime;
  int leftSubtreeSize;
  SpecialBST *left;
  SpecialBST *right;

  SpecialBST(int value, int idx, int numSmallerAtInsertTime) {
    this->value = value;
    this->idx = idx;
    this->numSmallerAtInsertTime = numSmallerAtInsertTime;
    leftSubtreeSize = 0;
    left = NULL;
    right = NULL;
  }

  void insert(int value, int idx, int numSmallerAtInsertTime = 0) {
    if (value < this->value) {
      leftSubtreeSize++;
      if (left == NULL) {
        left = new SpecialBST(value, idx, numSmallerAtInsertTime);
      } else {
        left->insert(value, idx, numSmallerAtInsertTime);
      }
    } else {
      numSmallerAtInsertTime += leftSubtreeSize;
      if (value > this->value)
        numSmallerAtInsertTime++;
      if (right == NULL) {
        right = new SpecialBST(value, idx, numSmallerAtInsertTime);
      } else {
        right->insert(value, idx, numSmallerAtInsertTime);
      }
    }
  }
};

void getRightSmallerCounts(SpecialBST *bst, vector<int> &rightSmallerCounts);

// Average case: when the created BST is balanced
// O(nlog(n)) time | O(n) space - where n is the length of the array
// ---
// Worst case: when the the created BST is like a linked list
// O(n^2) time | O(n) space
vector<int> rightSmallerThan(vector<int> array) {
  if (array.size() == 0)
    return {};

  int lastIdx = array.size() - 1;
  SpecialBST *bst = new SpecialBST(array[lastIdx], lastIdx, 0);
  for (int i = array.size() - 2; i >= 0; i--) {
    bst->insert(array[i], i);
  }

  vector<int> rightSmallerCounts = array;
  getRightSmallerCounts(bst, rightSmallerCounts);
  return rightSmallerCounts;
}

void getRightSmallerCounts(SpecialBST *bst, vector<int> &rightSmallerCounts) {
  if (bst == NULL)
    return;
  rightSmallerCounts[bst->idx] = bst->numSmallerAtInsertTime;
  getRightSmallerCounts(bst->left, rightSmallerCounts);
  getRightSmallerCounts(bst->right, rightSmallerCounts);
}
```