Solution 1   Solution 2   Solution 3

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

// O(n) time | O(n) space - where n is the length of the array
function minHeightBst(array) {
  return constructMinHeightBst(array, null, 0, array.length - 1);
}

function constructMinHeightBst(array, bst, startIdx, endIdx) {
  if (endIdx < startIdx) return;
  const midIdx = Math.floor((startIdx + endIdx) / 2);
  const newBstNode = new BST(array[midIdx]);
  if (bst === null) {
    bst = newBstNode;
  } else {
    if (array[midIdx] < bst.value) {
      bst.left = newBstNode;
      bst = bst.left;
    } else {
      bst.right = newBstNode;
      bst = bst.right;
    }
  }
  constructMinHeightBst(array, bst, startIdx, midIdx - 1);
  constructMinHeightBst(array, bst, midIdx + 1, endIdx);
  return bst;
}

class BST {
  constructor(value) {
    this.value = value;
    this.left = null;
    this.right = null;
  }

  // We don't use this method for this solution.
  insert(value) {
    if (value < this.value) {
      if (this.left === null) {
        this.left = new BST(value);
      } else {
        this.left.insert(value);
      }
    } else {
      if (this.right === null) {
        this.right = new BST(value);
      } else {
        this.right.insert(value);
      }
    }
  }
}

exports.minHeightBst = minHeightBst;
```