4.3: Create Binary Tree

September 7, 2017

Question

Given an array sorted in increasing order, return a binary tree such that the left and right subtrees have the same number of nodes or the number of nodes differs by at most 1.

Explanation and Algorithm

This is the algorithm that should be used:

- Insert into the tree the middle element of the array
- Insert the left subarray of elements into the left subtree
- Insert the right subarray of elements into the right subtree
- Recurse

Hints

1. Where in the array should you start? (Beginning, middle, end?)
2. Where in the array should you start if you wanted to ensure that the left and right subtrees were equal?
3. Is recursion useful for solving this problem?
4. Try writing out by hand the process your code should use to see if it will be effective. Use sample array [1, 2, 3, 4, 5, 6, 7]

Code

```
public static TreeNode addToTree(int arr[], int start, int end){
   if (end < start) {
      return null;
   }
   int mid = (start + end) / 2;
   TreeNode n = new TreeNode(arr[mid]);
   n.left = addToTree(arr, start, mid - 1);
   n.right = addToTree(arr, mid + 1, end);
   return n;
}

public static TreeNode createMinimalBST(int array[]) {
   return addToTree(array, 0, array.length - 1);
}
```

Big O analysis

O(n): n is the number of items in the array; each item is viewed once and a node is created from it and added to the tree

Sources

Question, answer and other material taken from Cracking the Coding Interview 6th edition by Gayle Laakmann McDowell.