[LeetCode] Unique Binary Search Trees I and II (Java)

July 29, 2014 by decoet

> ***Unique Binary Search Trees***
>
> *Given n, how many structurally unique **BST's** (binary search trees) that store values 1…n?*

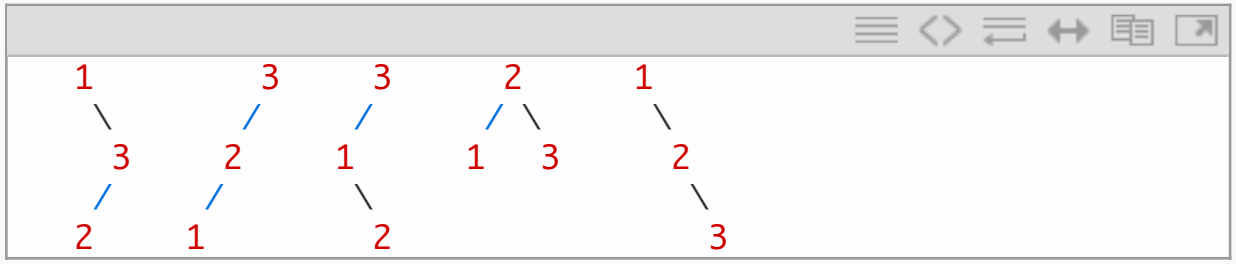
Search …

Recent Posts

[LeetCode] Queue Reconstruction by Height

[LeetCode] Median of Two Sorted Arrays (More elegant solution)

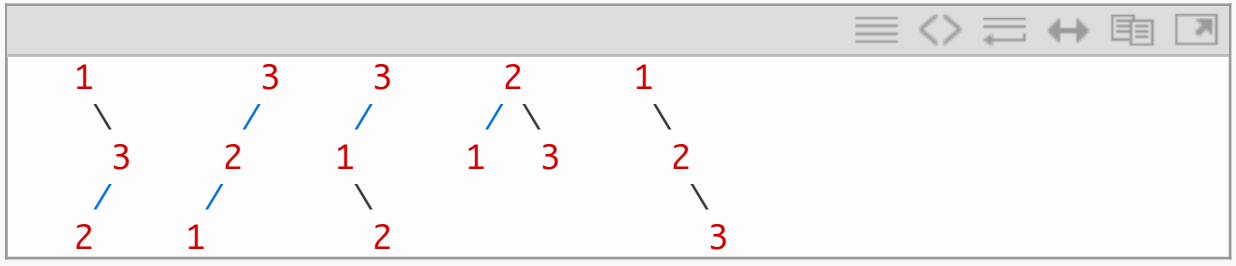
For example,
Given n = 3, there are a total of 5 unique BST's.



Unique Binary Search Trees II

*Given n, generate all structurally unique **BST's** (binary search trees) that store values 1…n.*

For example,
Given n = 3, your program should return all 5 unique BST's shown below.



Reading Notes for [All Abroad the Databus! – Linkedin's Scalable Consistent Change Data Capture Platform]

Reading Notes for [On Brewing Fresh Espresso: LinkedIn's Distributed Data Serving Platform]

Reading Notes for [Kafka, a Distributed Messaging System for Log Processing]

Recent Comments

[LeetCode] Median of Two Sorted Arrays

Analysis

In “Unique Binary Search Trees”, we are only required to output the number of the trees. We know that all nodes in the left subtree are smaller than the root. And all nodes in the right subtree are larger than the root. For a integer n, we have n options to be the root. In these options, assuming i is the value that we choose to be the root. The value in left subtree are from 1 to i – 1, and the values in right subtree are from i + 1 to n. If 1 to i – 1 can form p different trees, and i + 1 to n can form q different trees, then we will have p * q trees when i is the root. In fact, the number of different trees depends on how many number to form the tree.

We can use an array to save the number of different trees that n integers can form. We fill the array from bottom to up, starting from 0 to n.

In “Unique Binary Search Trees II”, we need to generate all trees. The algorithm has the same idea. But we don’t just return the numbers. We return the trees that n integers can from. Then we use a nested for-loop to go through every possible combinations of left tree and right tree for a given root. We will do it recursively because it’s the same for the left tree and right tree of root.

Code

Unique Binary Search Trees

```java
public class Solution {
    public int numTrees(int n) {
        int[] N = new int[n + 1];
        N[0] = 1;
        for (int i = 1; i <= n; i++) {
            for (int j = 0; j < i; j++) {
                N[i] += N[j] * N[i - j - 1];
            }
        }
        return N[n];
    }
}
```

Unique Binary Search Trees II

```java
public class Solution {
    public List<TreeNode> generateTrees(int n) {
        return generateTrees(1, n);
    }

    public List<TreeNode> generateTrees(int start, int end) {
        List<TreeNode> list = new LinkedList<>();
        if (start > end) {
            list.add(null);
            return list;
        }
        for (int i = start; i <= end; i++) {
            List<TreeNode> lefts = generateTrees(start, i - 1);
            List<TreeNode> rights = generateTrees(i + 1, end);
            for (TreeNode left : lefts) {
```

therasablog on [LeetCode] Rotate List (Java)

Archives

October 2016
September 2016
June 2016
March 2016
August 2015
August 2014
July 2014
June 2014
April 2014
March 2014
February 2014
November 2013
October 2013

```
                for (TreeNode right : rights) {
                    TreeNode node = new TreeNode(i);
                    node.left = left;
                    node.right = right;
                    list.add(node);
                }
            }
        }
        return list;
    }
}
```

Complexity

The complexity of the first problem is $O(n^2)$.

Programming

< [LeetCode] Decode Ways (Java)

> [LeetCode] Validate Binary Search Tree (Java)

1 Comment Life in Code

Recommend **Share**

September 2013

Categories

Life

Programming

System Design

Meta

Log in

Entries RSS

Comments RSS

WordPress.org

Join the discussion…

k • 2 years ago

That was a terrible analysis.

• Reply • Share ›

ALSO ON **LIFE IN CODE**

About Me

7 comments • 3 years ago•

Luye He — 大神太牛逼了交个朋友呗

[LeetCode]

2 comments • 2

random
remove

[LeetCode] Palindrome Partitioning I and II(Java)

2 comments • 2 years ago•

Truong Khanh Nguyen — This is my discussion and Java implementation …

[LeetCode]

1 comment • 2 y

Apoorv
work wit

Loading [Contrib]/a11y/accessibility-menu.js