

96. Unique Binary Search Trees

Medium,
DP, Tree.

Given n , how many structurally unique BST's (binary search trees) that store values $1 \dots n$?

Example:

Input: 3
Output: 5
Explanation:
Given $n = 3$, there are a total of 5 unique BST's:

```
  1   3  3  2   1
  \  /  /  /\  \
   3 2  1  1 3  2
  /  /  \      \
 2  1   2       3
```

解法

java

可以用dfs进行搜索。

```
class Solution {
    public int numTrees(int n) {
        return dfs(1, n);
    }

    private int dfs(int start, int end) {
        if (start >= end) {
            return 1;
        }
    }
}
```

```

int count = 0;
for (int i = start; i <= end; i++) {
    int left = dfs(start, i - 1);
    int right = dfs(i + 1, end);
    count += left * right;
}
return count;
}
}

```

java

用dp。

就跟斐波那契数列一样，我们把n = 0 时赋为1，因为空树也算一种二叉搜索树，那么n = 1 时的情况可以看做是其左子树个数乘以右子树的个数，左右字数都是空树，所以1乘1还是1。那么n = 2时，由于1和2都可以为跟，分别算出来，再把它们加起来即可。n = 2的情况可由下面式子算出：

$dp[2] = dp[0] * dp[1]$ (1为根的情况)
 $+ dp[1] * dp[0]$ (2为根的情况)

同理可写出 n = 3 的计算方法：

$dp[3] = dp[0] * dp[2]$ (1为根的情况)
 $+ dp[1] * dp[1]$ (2为根的情况)
 $+ dp[2] * dp[0]$ (3为根的情况)

由此可以得出卡特兰数列的递推式为：

$$C_0 = 1 \quad \text{and} \quad C_{n+1} = \sum_{i=0}^n C_i C_{n-i} \quad \text{for } n \geq 0.$$

我们根据以上的分析，可以写出代码如下：

```

class Solution {
    public int numTrees(int n) {
        if(n==1) return 1;
        int[] dp = new int[n+1];
        dp[0]=1;
        dp[1]=1;
        for(int i=2; i<=n; i++){
            for(int j=0; j<i; j++){
                dp[i] += dp[j]* dp[i-j-1];
            }
        }
    }
}

```

```
    }  
  }  
  return dp[n];  
}  
}
```