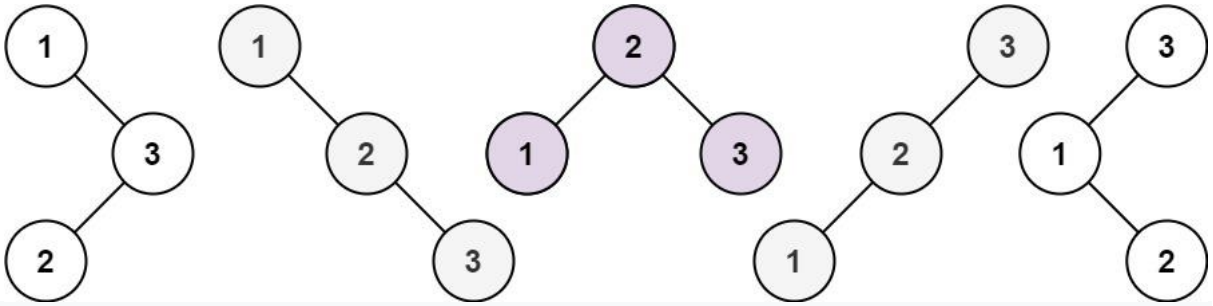
Given an integer `n`, return *the number of structurally unique* ***BST****'s (binary search trees) which has exactly* `n` *nodes of unique values from* `1` *to* `n`.

Example 1:




```
Input: n = 3

Output: 5
```

Example 2:

```
Input: n = 1

Output: 1
```

Constraints:

- `1 <= n <= 19`

SOLUTION

```
class Solution {

    public int numTrees(int n) {

        int[] count = new int[n + 1];


            count[0] = 1;

            count[1] = 1;
```

```
        for (int i = 2; i <= n; i++) {
            for (int j = 0; j <= i - 1; j++) {
                count[i] = count[i] + count[j] * count[i - j - 1];
            }
        }

        return count[n];

    }
```

OUTPUT