https://course.acciojob.com/idle?question=fd205fb2-7875-4269-9eba-1d4f5999042c

MEDIUM

Max Score: 40 Points

# Unique Binary Search Trees

**You are given n nodes with Unique values from 1 to n. Find the number of structurally unique BSTs.**

## Input Format

**First Line of input contains an integer N.**

## Output Format

**Print number of unique BSTs.**

## Example 1

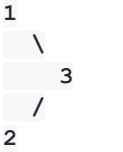
Input

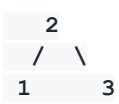
```
3
```

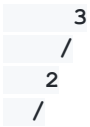
Output

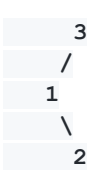
```
5
```

Explanation









## Example 2

Input

Output

Explanation

Only 1 tree possible.

## Constraints

**1 <= N <=20**

Topic Tags

Recursion

BST

DP

# My code

```
// in java
import java.util.*;

class Solution {
public static int numTrees(int n) {
  int [] G = new int[n+1];
  G[0] = G[1] = 1;

  for(int i=2; i<=n; ++i) {
    for(int j=1; j<=i; ++j) {
      G[i] += G[j-1] * G[i-j];
    }
  }
  return G[n];
}
        // static int fun(int i,int n)
        // {
        //      int ans=1;
        //      while(i>n)
        //              {
        //                      ans=ans*i--;
```

```
        //              }
        //      return ans;
        // }
      public static int uniqueBST(int n) {
       // Write your code here
              //  return fun(2*n,n)/fun(n+1,0);
                return numTrees(n);
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N= sc.nextInt();
        Solution Obj = new Solution();
        System.out.println(Obj.uniqueBST(N));

    }
}

// public int numTrees(int n) {
//   int [] G = new int[n+1];
//   G[0] = G[1] = 1;

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

```
// }
```