Number of Binary Search Tree

HackerRank

A binary tree is a tree which is characterized by any of the following properties:

1. It can be empty (null).
2. It can contain a root node which contain some value and two subtree, left subtree and right subtree, which are also binary tree.

A binary tree is a binary search tree (BST) if all the non-empty nodes follows both two properties:

1. If node has a left subtree, then all the values in its left subtree are smaller than the value of the current node.
2. If node has a right subtree, then all the value in its right subtree are greater than the value of the current node.

You are given *N* nodes, each having unique value ranging from `[1, N]`, how many different binary search tree can be created using all of them.

Input
First line will contain an integer, *T*, number of test cases. Then *T* lines follow, where each line represent a test case. Each test case consists a single integer, *N*, where *N* is the number of nodes in the binary search tree.

Output
For each test case, find the number of different binary search trees that can be created using these nodes. Print the answer modulo (10^8+7).

Constraints
1 <= *T* <= 1000
1 <= *N* <= 1000

Sample Input

```
5
1
2
3
4
100
```

Sample Output

```
1
2
5
14
25666077
```

Explanation

Test Case #1: We have only one tree.

```
1
```

Test Case #2: Two trees can be created using two nodes.

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

Test Case #3:

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