AlgoExpert | Quad Layout | Java | 12px | Sublime | Monokai | 00:00:00

Prompt | Scratchpad | Our Solution(s) | Video Explanation | Run Code

Solution 1 | Solution 2 | Solution 3

```java
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

import java.util.*;

class Program {
  // O(n^2) time | O(n) space
  public static int numberOfBinaryTreeTopologies(int n) {
    List<Integer> cache = new ArrayList<Integer>();
    cache.add(1);
    for (int m = 1; m < n + 1; m++) {
      int numberOfTrees = 0;
      for (int leftTreeSize = 0; leftTreeSize < m; leftTreeSize++) {
        int rightTreeSize = m - 1 - leftTreeSize;
        int numberOfLeftTrees = cache.get(leftTreeSize);
        int numberOfRightTrees = cache.get(rightTreeSize);
        numberOfTrees += numberOfLeftTrees * numberOfRightTrees;
      }
      cache.add(numberOfTrees);
    }
    return cache.get(n);
  }
}
```