

# Leetcode Solutions - Easy Medium Hard

I have shared solutions for leetcode problems of easy, medium and hard levels with step by step explanations for solving the problems.

## 96. Unique Binary Search Trees



April 18, 2021

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:

- $1 \leq n \leq 19$

### Important observation for solving the problem:

- If we keep  $K$  as our root, then, all numbers lower than  $K$  will come in left subtree and all numbers greater than  $1$  will come in right subtree.
- Also, if we have  $X$  ways of forming left subtree and  $Y$  ways of forming right subtree, then, total number of trees rooted at  $K$  can be calculated as :  $\text{ways}(K) = \text{ways}(X) * \text{ways}(Y)$ .
- The **base case** here is, if we have less than  $1$  nodes left, then we have only **one** way of forming such subtree.

### Solving using recursion (TLE):

- We have nodes from range  $[1, n]$ . We will make it as  $[0, n - 1]$  for our convenience.
- Now suppose we take node 5 as our current root, then, the left subtree will contain nodes  $[0, 4]$  and right subtree will contain nodes  $[6, n - 1]$ .
- We will solve this recursively, and when we reach our base case, we will return 1 as our answer.

### Code using recursion:

```
class Solution {
    public int numTrees(int n) {
        return recur(0, n - 1);
    }

    int recur(int l, int r) {
        // our base case
        if(l >= r) return 1;

        int ways = 0;
        // choosing i as the root node of current subtree
        for(int i = l; i <= r; i++) {
            ways = ways + recur(l, i - 1) * recur(i + 1, r);
        }

        return ways;
    }
}
```

### Caching the subproblems (TLE to Accepted):

- Now the above code gives TLE.
- This is because we are computing subproblems again and again.
- So we will just cache the results in an array or map, as we want to do.

### Code using DP:

```
class Solution {
```

```
// array to store results of subproblems
Integer[][] dp;

public int numTrees(int n) {
    dp = new Integer[n][n];
    return recur(0, n - 1);
}

int recur(int l, int r) {
    if(l >= r) return 1;

    // checking if we have already solved this subproblem
    // if yes, we will return it's answer
    if(dp[l][r] != null)
        return dp[l][r];
    int ways = 0;
    for(int i = l; i <= r; i++) {
        ways = ways + recur(l, i - 1) * recur(i + 1, r);
    }

    // storing the result of the current subproblem
    // so that it can be used later on
    dp[l][r] = ways;

    return ways;
}
```

To leave a comment. click the button below to sign in with Google.

SIGN IN WITH GOOGLE

#### Popular posts from this blog

### 123. Best Time to Buy and Sell Stock III

April 19, 2021

Blog You are given an array prices where prices[i] is the price of a given stock on the i<sup>th</sup> day. Find the maximum profit you can achieve. You may complete at most two transactions . Note: You may not engage in multiple transactions simultaneously (i.e., you must sell the stock before you can buy again).

[READ MORE](#)

---

## 1799. Maximize Score After N Operations (Leetcode)

March 21, 2021

Blog You are given nums , an array of positive integers of size  $2 * n$  . You must perform n operations on this array. In the i<sup>th</sup> operation (1-indexed) , you will: Choose two elements, x and y . Receive a score of  $i * \gcd(x, y)$  . Remove x and y from nums . Return the maximum score you can achieve.

[READ MORE](#)

---

## 1824. Minimum Sideway Jumps

April 11, 2021

Blog There is a 3 lane road of length n that consists of n + 1 points labeled from 0 to n . A frog starts at point 0 in the second lane and wants to jump to point n . However, there could be obstacles along the way. You are given an array obstacles of length n + 1 where  $obstacles[i]$  is the lane of the obstacle at point i .

[READ MORE](#)

 Powered by Blogger

Theme images by [Michael Elkan](#)

