<https://www.codingninjas.com/codestudio/problems/problem-name-boolean-evaluation_1214650?source=youtube&campaign=striver_dp_videos&utm_source=youtube&utm_medium=affiliate&utm_campaign=striver_dp_videos&leftPanelTab=0>

f(i, j, 1) - number of true in i to j block.

f(i, j, 0) - number of false in i to j block

**Approach:** f(i, j, isTrue)

1. Start with the entire block/ array.
2. Try taking every operand as partition one by one
3. Add all the ways and return

**Recursive soln:**

1. Take entire block by passing i = 0, j= n-1
2. Try all possible operand as starting partition = i+1 to j(by 2 jumps).
3. Make lt, lf, rt, rf that will give left’s/right’s, true/false.

**AND** => ways(for true) = lt\*rt; //TT

**OR** => ways(--) = lt\*rt + lt\*rf + lf\*rt; //TT + TF + FT

**XOR** => ways(--) = lt\*rf + lf\*rt; //TF + FT

Like that count for false also and acc to function true of false what is needed add up them.

1. Keep adding ways and return that..

**optional**

**Tabulation(bottom-up):** i = n-1 to 0 and j = i to n

1. Initialize dp[n+1][n+1][2] with 0 to counter base case.
2. Use the same recurrence relation to build dp.
3. Return dp[0][n-1][1]