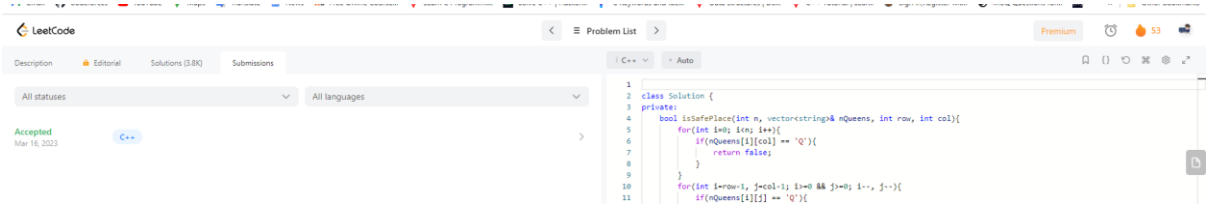
Code-

```
class Solution {
private:
    bool isSafePlace(int n, vector<string>& nQueens, int row, int col){
        for(int i=0; i<n; i++){
            if(nQueens[i][col] == 'Q'){
                return false;
            }
        }
        for(int i=row-1, j=col-1; i>=0 && j>=0; i--, j--){
            if(nQueens[i][j] == 'Q'){
                return false;
            }
        }
        for(int i=row-1, j=col+1; i>=0 && j<n; i--, j++){
            if(nQueens[i][j] == 'Q'){
                return false;
            }
        }
        return true;
    }
    void solveNQueens(int n, vector<vector<string>>& output, vector<string>&
nQueens, int row){
        if(row == n){
            output.push_back(nQueens);
            return;
        }
        for(int col=0; col<n; col++){
            if(isSafePlace(n, nQueens, row, col)){
                nQueens[row][col] = 'Q';
                solveNQueens(n, output, nQueens, row+1);
                nQueens[row][col] = '.';
            }
        }
    }
public:
    vector<vector<string>> solveNQueens(int n) {
        vector<vector<string>> output;
        vector<string> nQueens(n , string(n, '.'));
        solveNQueens(n, output, nQueens, 0);
        return output;
    }
};
```

Time complexity –O(N!)

Space complexity –O(N^2)


LeetCode
Problem List
Premium
53
Description
Editorial
Solutions (3.8K)
Submissions
All statuses
All languages
Accepted
Mar 16, 2023
C++
C++
Auto
class Solution {
private:
bool isSafePlace(int n, vector<string>& nQueens, int row, int col){
for(int i=0; i<n; i++){
if(nQueens[i][col] == 'Q'){
return false;
}
}
for(int i=row-1, j=col-1; i>=0 && j>=0; i--, j--){
if(nQueens[i][j] == 'Q'){