**SOLUTION**

class Solution {

public:

Solution(){

ios::sync\_with\_stdio(false);

std::cin.tie(nullptr);

std::cout.tie(nullptr);

}

int maximalSquare(vector<vector<char>>& matrix) {

int r=matrix.size();

if(r==0)

return 0;

int c=matrix[0].size();

if(c==0)

return 0;

int square=0;

vector<vector<int>> dp(r,vector<int>(c,0));

for(int i=0;i<r;i++){

for(int j=0;j<c;j++){

if(i==0 || j==0){

if(matrix[i][j]=='1'){

dp[i][j]=1;

square=max(square,dp[i][j]);

}

}

else{

if(matrix[i][j]=='1'){

dp[i][j]=1+min({dp[i-1][j-1],dp[i-1][j],dp[i][j-1]});

square=max(square,dp[i][j]);

}

}

}

}

return square\*square;

}

};

**TIME COMPLEXITY: O(N^2)**

**SPACE COMPLEXITY: O(N^2)**