Day 60

Code-

class Solution {

public:

int m,n;

vector<vector<int>> dp;

int isSquare(vector<vector<char>>& mat, int i=0, int j=0)

{

if(i+1==m || j+1==n || mat[i][j]=='0')

return 1;

if(mat[i][j]=='0')

return 1;

if(i+1!=m and j+1!=n and mat[i+1][j+1]=='0')

return 1;

if(i+1!=m and mat[i+1][j]=='0')

return 1;

if(j+1!=n and mat[i][j+1]=='0')

return 1;

if(dp[i][j]!=-1)

return dp[i][j];

int a = isSquare(mat,i+1,j);

int b = isSquare(mat,i,j+1);

int c = isSquare(mat,i+1,j+1);

return dp[i][j] = 1 + min({a,b,c});

}

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

m = matrix.size();

n = matrix[0].size();

dp.resize(m,vector<int>(n,-1));

int mx=0;

for(int i=0;i<m;i++)

{

for(int j=0;j<n;j++)

{

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

int val = pow(isSquare(matrix,i,j),2);

mx = max(mx,val);

}

}

}

return mx;

}

};};Time Complexity : O(m \*log(n))  
Space Complexity: O(n )

