1. Maximal Square

Given a 2D binary matrix filled with 0’s and 1’s, find the largest square containing only 1’s and return its area.

**Example:**

Input:   
  
1 0 1 0 0  
1 0 1 1 1  
1 1 1 1 1  
1 0 0 1 0  
  
Output: 4

**解** 动态规划，设表示下标为(i, j)的左上角区域的最大正方形面积，当时：

class Solution {  
public:  
 int maximalSquare(vector<vector<char>>& matrix) {  
 if(matrix.size() == 0)return 0;  
 int m = matrix.size(), n = matrix[0].size();  
 int \*dp = new int[n+1];  
 memset(dp, 0, (n+1)\*sizeof(int));  
 int ans = 0, prev = 0;  
 for(int i = 1; i <= m; ++i){  
 prev = dp[0];  
 for(int j = 1; j <= n; ++j){  
 int tmp = dp[j];  
 if(matrix[i-1][j-1] == '1'){  
 dp[j]=min(dp[j], min(prev, dp[j-1]))+1;  
 ans = max(ans, dp[j]);  
 }else{  
 dp[j] = 0;  
 }  
 prev = tmp;  
 }  
 }  
 return ans\*ans;  
   
 }  
};