/\*

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

For example, given the following matrix:

1 0 1 0 0

1 0 1 1 1

1 1 1 1 1

1 0 0 1 0

Return 6.

way-1:运用上一道题的 Largest Rectangle in Histogram 。

把矩阵抽象成一个个Histogram。

\*/

class Solution {

public:

int largestRectangleArea(vector<int>& height)

{

int result=0;

stack<int> stk;

for(int i=0;i<height.size();i++)

{

if(stk.empty() || stk.top()<=height[i])

stk.push(height[i]);

else

{

int count=0;

while(!stk.empty() && stk.top()>height[i])

{

count++;

result=max(result,stk.top()\*count);

stk.pop();

}

while(count--)

stk.push(height[i]);

stk.push(height[i]);

}

}

int count=1;

while(!stk.empty())

{

result=max(result,stk.top()\*count);

count++;

stk.pop();

}

return result;

}

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

{

if(matrix.size()==0)

return 0;

int row=matrix.size();

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

//chat to int

vector<vector<int>> result;

vector<int> m1;

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

{

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

{

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

m1.push\_back(1);

else

m1.push\_back(0);

}

result.push\_back(m1);

m1.clear();

}

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

{

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

if(result[i][j]==1 && j!=0)

result[i][j]=result[i][j-1]+1;

}

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

{

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

cout<<result[i][j];

cout<<endl;

}

cout<<endl;

vector<vector<int>> jxn;

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

{

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

{

if(result[j][i]!=0)

m1.push\_back(result[j][i]);

else if(result[j][i]==0 || j==row-1)

{

if(m1.size()!=0)

{

jxn.push\_back(m1);

m1.clear();

}

}

}

if(m1.size()!=0)

{

jxn.push\_back(m1);

m1.clear();

}

}

for(int i=0;i<jxn.size();i++)

{

for(int j=0;j<jxn[i].size();j++)

cout<<jxn[i][j];

cout<<endl;

}

int maxa=0;

for(int i=0;i<jxn.size();i++)

maxa=max(maxa,largestRectangleArea(jxn[i]));

return max;

}

};