Program Name/Purpose- Find the maximum area sub-matrix that has only 1's. Note that we are interested in finding the **maximum sub-matrix** and not **maximum sub-square**.(http://www.geeksforgeeks.org/maximum-size-sub-matrix-with-all-1s-in-a-binary-matrix/)

Programming Paradigm Used- Dynamic Programming

Data Structures Used- Stack to get the maximum area in histogram

Explanation-

Suppose we have an input matrix like-

Input Matrix

0	1	1	0	1
1	1	0	1	0
0	1	1	1	0
1	1	1	1	1
1	1	1	1	1
0	0	0	0	0

Now, we want to get the maximum sub-matrix with all 1's.
If we consider only **square sub-matrix** , then the maximum area found will be- **9 sq. unit**, whereas actually the maximum sub-matrix is a **rectangle** having an area of **10 sq.unit.**

For finding maximum sub-square matrix you can visit- http://www.geeksforgeeks.org/maximum-size-sub-matrix-with-all-1s-in-a-binary-matrix/

See the below figure for clarity-

Input Matrix

0	1	1	0	1
1	1	0	1	0
0	1	1	1	0
1	1	1	1	1
1	1	1	1	1
0	0	0	0	0

Area of maximum sub-square matrix is - 9 sq unit.

Input Matrix

0	1	1	0	1
1	1	0	1	0
0	1	1	1	0
1	1	1	1	1
1	1	1	1	1
0	0	0	0	0

Area of maximum sub-matrix is - 10 sq unit which is not a square, but a rectangle.

So the algorithm would be to compute an auxiliary matrix using the input matrix by using dynamic programming.

This auxiliary matrix , at each row, stores the histograms heights originating from that row.

Input Matrix

0	1	1	0	1
1	1	0	1	0
0	1	1	1	0
1	1	1	1	1
1	1	1	1	1
0	0	0	0	0

Conversion from input to auxiliary matrix using the Dynamic Programming relation-

```
if(Aux[i][j] == 1)
        Aux[i][j] = Aux[i-1][j] + 1;
```

Auxiliary Matrix

0	1	1	0	1
1	2	0	1	0
0	3	1	2	0
1	4	2	3	1
2	5	3	4	2
0	0	0	0	0

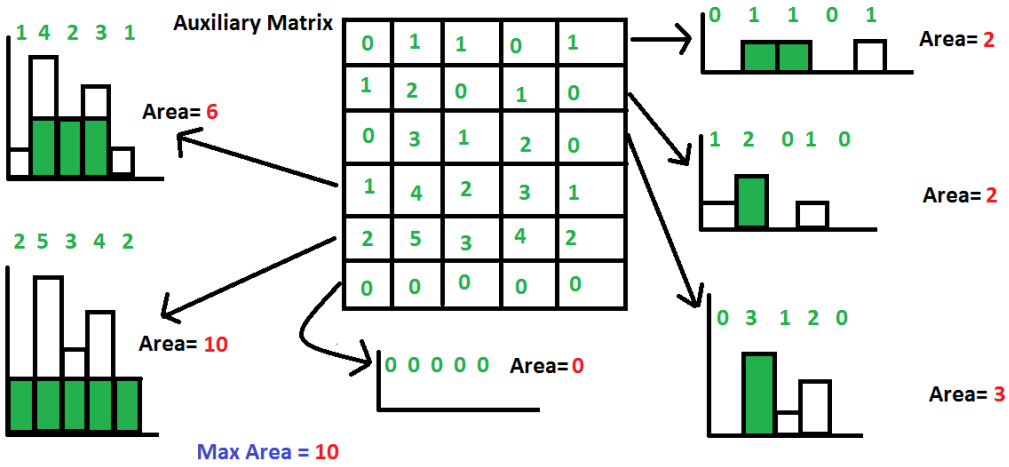
Now, by using auxiliary matrix(at the right side on the above figure), we find the maximum area of histograms starting from each row.The maximum of these area is our required answer.

For more information on finding the area of histogram, visit-
http://www.geeksforgeeks.org/largest-rectangle-under-histogram/

Why this works ?

The area we find at each row gives the maximum sub-matrix containing only 1's ending at that row. Now the maximum of all these areas gives the maximum sub-matrix containing only 1's in the whole input matrix.

The algorithm is shown in the below figure clearly-



Since , the maximum area is **10 sq.unit** which is our required answer.

Time Complexity-

We are calculating histogram ending at **each row**. And calculating one such area takes **O(column)** time. Hence the overall time complexity is – **O(rows*columns).**

Space Complexity-

We create an auxiliary table, so **O(rows*columns) space.**