

[geeksforgeeks.org](https://www.geeksforgeeks.org)

# Maximum size rectangle binary sub-matrix with all 1s - GeeksforGeeks

*GeeksforGeeks*

4-5 minutes

---

Given a binary matrix, find the maximum size rectangle binary-sub-matrix with all 1's.

## Example:

### Input:

```
0 1 1 0
1 1 1 1
1 1 1 1
1 1 0 0
```

### Output :

8

### Explanation :

The largest rectangle with only 1's is from  
(1, 0) to (2, 3) which is

```
1 1 1 1
1 1 1 1
```

### Input:

```
0 1 1
1 1 1
```

```
0 1 1
```

**Output:**

**6**

**Explanation :**

The largest rectangle with only 1's is from

(0, 1) to (2, 2) which is

```
1 1
```

```
1 1
```

```
1 1
```

*There is already an algorithm discussed a [dynamic programming based solution for finding](#) the [largest square with 1s](#).*

**Approach:**

In this post, an interesting method is discussed that uses [largest rectangle under histogram](#) as a subroutine.

If the height of bars of the histogram is given then the largest area of the histogram can be found. This way in each row, the largest area of bars of the histogram can be found. To get the largest rectangle full of 1's, update the next row with the previous row and find the largest area under the histogram, i.e. consider each 1's as filled squares and 0's with an empty square and consider each row as the base.

**Illustration:**

**Input :**

```
0 1 1 0
```

```
1 1 1 1
```

```
1 1 1 1
```

```
1 1 0 0
```

**Step 1:**

0 1 1 0 maximum area = 2

### Step 2:

row 1 1 2 2 1 area = 4, maximum area becomes 4

row 2 2 3 3 2 area = 8, maximum area becomes 8

row 3 3 4 0 0 area = 6, maximum area remains 8

### Algorithm:

1. Run a loop to traverse through the rows.
2. Now If the current row is not the first row then update the row as follows, if  $\text{matrix}[i][j]$  is not zero then  $\text{matrix}[i][j] = \text{matrix}[i-1][j] + \text{matrix}[i][j]$ .
3. Find the maximum rectangular area under the histogram, consider the  $i$ th row as heights of bars of a histogram. This can be calculated as given in this article [Largest Rectangular Area in a Histogram](#)
4. Do the previous two steps for all rows and print the maximum area of all the rows.

**Note:** It is strongly recommended to refer to [this](#) post first as most of the code is taken from there.

### Implementation

- C++
- Java
- Python3
- C#
- Javascript

### C++

```
#include <bits/stdc++.h>

using namespace std;

#define R 4

#define C 4

int maxHist(int row[])
{
    stack<int> result;

    int top_val;

    int max_area = 0;

    int area = 0;

    int i = 0;

    while (i < C) {

        if (result.empty() || row[result.top()] <=
row[i])

            result.push(i++);

        else {

            top_val = row[result.top()];

            result.pop();

            area = top_val * i;

            if (!result.empty())

                area = top_val * (i - result.top() -
1);

            max_area = max(area, max_area);
        }
    }
}
```

```
        }
    }
    while (!result.empty()) {
        top_val = row[result.top()];
        result.pop();
        area = top_val * i;
        if (!result.empty())
            area = top_val * (i - result.top() - 1);
        max_area = max(area, max_area);
    }
    return max_area;
}

int maxRectangle(int A[][C])
{
    int result = maxHist(A[0]);
    for (int i = 1; i < R; i++) {
        for (int j = 0; j < C; j++)
            if (A[i][j])
                A[i][j] += A[i - 1][j];
        result = max(result, maxHist(A[i]));
    }
    return result;
}
```

```
}

int main()
{
    int A[][C] = {
        { 0, 1, 1, 0 },
        { 1, 1, 1, 1 },
        { 1, 1, 1, 1 },
        { 1, 1, 0, 0 },
    };

    cout << "Area of maximum rectangle is "
          << maxRectangle(A);

    return 0;
}
```

## Java

## Python3

## C#

## Javascript

## Output

Area of maximum rectangle is 8

## Complexity Analysis:

- **Time Complexity:**  $O(R \times C)$ .

Only one traversal of the matrix is required, so the time complexity is  $O(R \times C)$

- **Space Complexity:**  $O(C)$ .

Stack is required to store the columns, so space complexity is  $O(C)$

This article is contributed by Sanjiv Kumar. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to [review-team@geeksforgeeks.org](mailto:review-team@geeksforgeeks.org). See your article appearing on the GeeksforGeeks main page and help other Geeks.