代码

```cpp
// C++ program to find maximum rectangular area in linear time
#include<iostream>
#include<stack>
using namespace std;

// The main function to find the maximum rectangular area under given
// histogram with n bars
int getMaxArea(int hist[], int n)
{
    // Create an empty stack. The stack holds indexes of hist[] array
    // The bars stored in stack are always in increasing order of their
    // heights.
    stack<int> s;

    int max_area = 0; // Initalize max area
    int tp;  // To store top of stack
    int area_with_top; // To store area with top bar as the smallest bar

    // Run through all bars of given histogram
    int i = 0;
    while (i < n)
    {
        // If this bar is higher than the bar on top stack, push it to stack
        if (s.empty() || hist[s.top()] <= hist[i])
            s.push(i++);

        // If this bar is lower than top of stack, then calculate area of rectangle
        // with stack top as the smallest (or minimum height) bar. 'i' is
        // 'right index' for the top and element before top in stack is 'left index'
        else
        {
            tp = s.top();  // store the top index
            s.pop();  // pop the top

            // Calculate the area with hist[tp] stack as smallest bar
            area_with_top = hist[tp] * (s.empty() ? i : i - s.top() - 1);

            // update max area, if needed
```

```
            if (max_area < area_with_top)
                max_area = area_with_top;
        }
    }

    // Now pop the remaining bars from stack and calculate area with every
    // popped bar as the smallest bar
    while (s.empty() == false)
    {
        tp = s.top();
        s.pop();
        area_with_top = hist[tp] * (s.empty() ? i : i - s.top() - 1);

        if (max_area < area_with_top)
            max_area = area_with_top;
    }

    return max_area;
}

// Driver program to test above function
int main()
{
    int hist[] = {6, 2, 5, 4, 5, 1, 6};
    int n = sizeof(hist)/sizeof(hist[0]);
    cout << "Maximum area is " << getMaxArea(hist, n);
    return 0;
}
```