

```
1 #include "../bits/stdc++.h"
2
3 // Longest Increasing Subsequencd
4 // verified: http://judge.u-aizu.ac.jp/onlinejudge/review.jsp?rid=3381664
5 int lis(std::vector<int> &a, int n)
6 {
7     const int INF = 1e9;
8     std::vector<int> dp(n, INF);
9     for (int i = 0; i < n; i++)
10     {
11         *lower_bound(dp.begin(), dp.end(), a[i]) = a[i];
12     }
13     return lower_bound(dp.begin(), dp.end(), INF) - dp.begin();
14 }
15
16 /**
17  * ヒストグラム h_1, h_2, ..., h_n が与えられる(各幅は1とする)
18  * 包含される長方形の内面積最大のものを求める
19  * L[i] := (j <= i && h_{j-1} < h_i) となるような最大の j
20  * R[i] := (j > i && h_j < h_i) となるような最大の j
21  */
22 template <typename T>
23 T maxRectangle(std::vector<T> &h)
24 {
25     int n = h.size();
26     std::vector<int> L(n), R(n);
27     std::vector<int> st(n);
28
29     int t = 0; // stack size
30     for (int i = 0; i < n; i++)
31     {
32         while (t > 0 && h[st[t - 1]] >= h[i])
33             t--;
34         L[i] = t == 0 ? 0 : (st[t - 1] + 1);
35         st[t++] = i;
36     }
37     t = 0;
38     for (int i = n - 1; i >= 0; i--)
39     {
40         while (t > 0 && h[st[t - 1]] >= h[i])
41             t--;
42         R[i] = t == 0 ? n : st[t - 1];
43         st[t++] = i;
44     }
45
46     T ret = 0;
47     for (int i = 0; i < n; i++)
48         ret = std::max(ret, h[i] * (R[i] - L[i]));
49     return ret;
50 }
51
```