Largest Rectangle ★

Your Largest Rectangle submission got 50.00 points. Share Tweet

Try the next challenge

Problem Submissions Leaderboard Editorial

Skyline Real Estate Developers is planning to demolish a number of old, unoccupied buildings and construct a shopping mall in their place. Your task is to find the largest solid area in which the mall can be constructed.

There are a number of buildings in a certain two-dimensional landscape. Each building has a height, given by $h[i] \text{ where } i \in [1,\ n]$. If you join k adjacent buildings, they will form a solid rectangle of area $k \times min(h[i], h[i+1], \ldots, h_[i+k-1])$.

Example

$h = [3, 2, 3]$

A rectangle of height $h = 2$ and length $k = 3$ can be constructed within the boundaries. The area formed is $h \cdot k = 2 \cdot 3 = 6$.

Function Description

Complete the function `largestRectangle` int the editor below. It should return an integer representing the largest rectangle that can be formed within the bounds of consecutive buildings.

largestRectangle has the following parameter(s):

- int h[n]: the building heights

Returns

- long: the area of the largest rectangle that can be formed within the bounds of consecutive buildings

Input Format

The first line contains n, the number of buildings.
The second line contains n space-separated integers, each the height of a building.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq h[i] \leq 10^6$

Sample Input

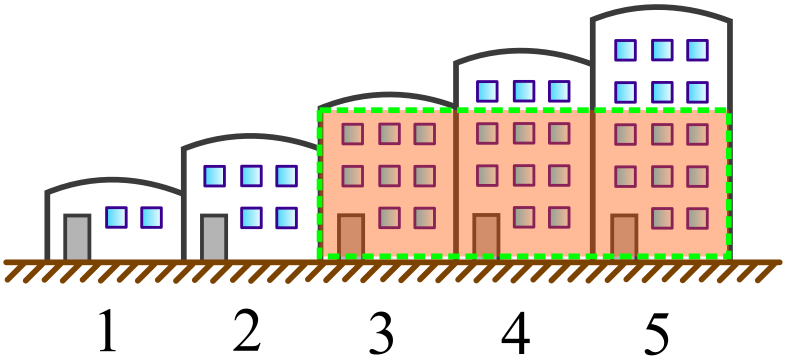
```
STDIN       Function
-----       --------
5           h[] size n = 5
1 2 3 4 5   h = [1, 2, 3, 4, 5]
```

Sample Output

```
9
```

Explanation

An illustration of the test case follows.



Change Theme Language C#

```
                var lastWidth = int.MaxValue;
                while(hStack.Count != 0 &&  hStack.Peek() > h[i])
                {
                    lastWidth = iStack.Peek();
                    var currentArea = (i - iStack.Pop()) * hStack.Pop();
                    max = Math.Max(max, currentArea);
                }
                if(hStack.Count == 0 || hStack.Peek() < h[i])
                {
                    hStack.Push(h[i]);
                    iStack.Push(Math.Min(lastWidth, i));
                }
            }
            return max;
    }

}

class Solution
{
    public static void Main(string[] args)
    {
        TextWriter textWriter = new StreamWriter(@System.Environment.GetEnvironmentVariable("OUTPUT_PATH"), true);

        int n = Convert.ToInt32(Console.ReadLine().Trim());

        List<int> h = Console.ReadLine().TrimEnd().Split(' ').ToList().Select(hTemp =>
```

Line: 47 Col: 24

Upload Code as File □ Test against custom input

Run Code Submit Code

You have earned 50.00 points!

These points will also count towards your progress in the Problem Solving Badge.

10% 990/2200

Problem Solving ★★★★★

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Earn a certificate in Problem Solving

Kudos on your progress! Take the HackerRank Skills Certification test and enrich your profile

Get Certified

- ✓ Test case 0
- ✓ Test case 1 🔒
- ✓ Test case 2 🔒
- ✓ Test case 3 🔒
- ✓ Test case 4 🔒
- ✓ Test case 5 🔒
- ✓ Test case 6 🔒

Compiler Message

```
Success
```

Input (stdin) Download

```
5
1 2 3 4 5
```

Expected Output Download

```
9
```