Codility_

Candidate Report: Anonymous

Check out Codility training tasks

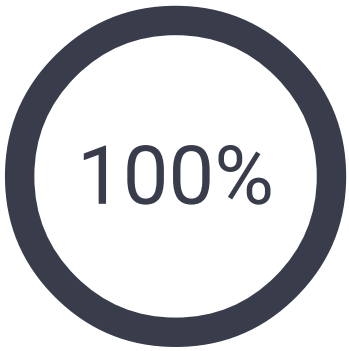
Test Name:

Summary Timeline Feedback

Tasks summary

Task	Time spent	Score
StoneWall Swift 4	55 min	100%

Total score



Tasks Details

Easy

1. **StoneWall**
Cover "Manhattan skyline" using the minimum number of rectangles.

Task Score	Correctness	Performance
100%	100%	100%

Task description

You are going to build a stone wall. The wall should be straight and N meters long, and its thickness should be constant; however, it should have different heights in different places. The height of the wall is specified by an array H of N positive integers. H[I] is the height of the wall from I to I+1 meters to the right of its left end. In particular, H[0] is the height of the wall's left end and H[N−1] is the height of the wall's right end.

The wall should be built of cuboid stone blocks (that is, all sides of such blocks are rectangular). Your task is to compute the minimum number of blocks needed to build the wall.

Write a function:

```
public func solution(_ H : inout [Int]) -> Int
```

that, given an array H of N positive integers specifying the height of the wall, returns the minimum number of blocks needed to build it.

For example, given array H containing N = 9 integers:

```
H[0] = 8    H[1] = 8    H[2] = 5
H[3] = 7    H[4] = 9    H[5] = 8
H[6] = 7    H[7] = 4    H[8] = 8
```

the function should return 7. The figure shows one possible arrangement of seven blocks.

Solution

Programming language used:	Swift 4
Total time used:	55 minutes
Effective time used:	55 minutes
Notes:	*not defined yet*

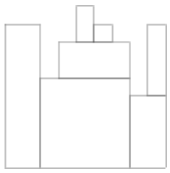
Task timeline

11:05:29 12:00:06

Code: 12:00:06 UTC, swift4, final, score: **100**

show code in pop-up

```
1  import Foundation
2  import Glibc
3
4  public struct Stack<T> {
5
6    /// Datastructure consisting of a generic item.
7    fileprivate var array = [T]()
8
```



Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [1..100,000];
- each element of array H is an integer within the range [1..1,000,000,000].


```
  /// The number of items in the stack.
  public var count: Int {
    return array.count
  }

  /// Verifies if the stack is empty.
  public var isEmpty: Bool {
    return array.isEmpty
  }

  /**
     Pushes an item to the top of the stack.

     - Parameter element: The item being pushed.
  */
  public mutating func push(_ element: T) {
    array.append(element)
  }

  /**
     Removes and returns the item at the top of the sta

     - Returns: The item at the top of the stack.
  */
  public mutating func pop() -> T? {
    return array.popLast()
  }

  /// Returns the item at the top of the stack.
  public var top: T? {
    return array.last
  }
}


public func solution(_ H : inout [Int]) -> Int {

    var stack = Stack<Int>()
    var counter = 0
    outer: for item in H {

        if let top = stack.top {
            if item < top {
                counter += 1
                stack.pop()
                inner: while let t = stack.top {
                    if item < t {
                        counter += 1
                        stack.pop()
                    }else if item == t {
                        continue outer
                    }else{
                        break inner
                    }
                }
                stack.push(item)
            }else if item > top {
                stack.push(item)
            }else{
                // if item == top
                // do nothing
            }
        }else{
            stack.push(item)
        }
    }

    return counter + stack.count
}
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: **O(N)**

expand all	Example tests	
▶	example	✔ OK
expand all	Correctness tests	
▶	simple1	✔ OK
▶	simple2	✔ OK
▶	simple3	✔ OK
▶	simple4	✔ OK
▶	boundary_cases	✔ OK
expand all	Performance tests	
▶	medium1	✔ OK
▶	medium2	✔ OK
▶	medium3	✔ OK
▶	medium4	✔ OK
▶	large_piramid	✔ OK
▶	large_increasing_decreasing	✔ OK
▶	large_up_to_20	✔ OK
▶	large_up_to_100	✔ OK
▶	large_max	✔ OK