Min Stack

Design a min stack class with the following methods:

* push(value) - Push an integer onto the stack.
* pop() - Removes and returns the element on top of the stack
* peek() - Return the top value in the stack (do not remove it)
* min() - Return the minimum value in the stack

Input: N/A  
Output: Instance of a min stack

# Example

Input: minStack.push(-3)  
 minStack.push(1)  
 minStack.push(5)  
 minStack.min()

Output: -3

# Constraints

Push, Pop, Peek, and Min must be performed in:  
 Time Complexity: O(1)  
 Auxiliary Space Complexity: O(1)

If interviewee asks, allow them to use an array or linked list or stack data structure

# Solution

1. Create two stacks using arrays (or dynamic arrays if java) called ’ values’ and ‘mins’
2. push(input)
   1. push the input into the ‘values’ array
   2. find lower of the last item in the ‘mins’ array and the input value
   3. push the lower of the two into the ‘mins’ array
3. pop()
   1. pop from the ‘min’ array, then pop and return the element from the ‘values’ array
4. peek() return the last item in the ‘values’ array
5. min(): return the last item in the ‘mins’ array

# Notes

N/A

# Resources

https://leetcode.com/problems/min-stack/