DAY-39

Design a stack that supports get_min () in O(1) time and O(1) extra space.

Approach:-

Define a variable minEle that stores current minimum element in the stack. now, the interaction part is, how to handle the case when minimum element is removed.

To handle this, we push "2x-minEle" into the stack instead of x so that previous minimum element can be retried using current minEle and its value stored in stack,

push (x):

Inserts n at the top of the stack.

→ if stack is empty, insert x into the stack and make minEle equal to n.

→ if stack is not empty, compare n with minEle. Two cases arise:

i) if n is greater than or equal to minEle, simply insert n.

ii) if n is less than minEle, insert (2*n - minEle) into the stack and make minEle equal to n.

pop ( ) :- removes an element from top of the stack. (y)

-> if y is greater than or equal to minEle , the minimum element in the stack is still minEle .

→ if y is less than minEle, the minimum now becomes (2*minEle - y), so update (minEle = 2*minEle - y).

[this is where we retrieve previous minimum from current minimum and its value in stack.

(*) points :-

-> stack doesn't hold actual value of an element if it is minimum so far.

-> Actual minimum