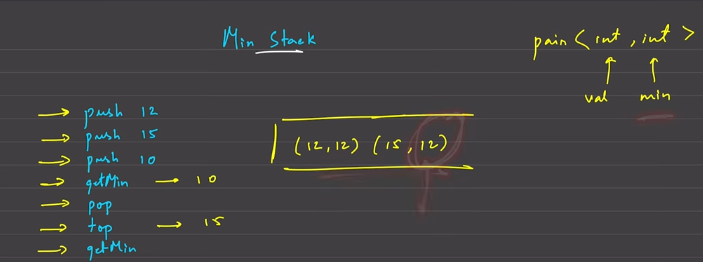
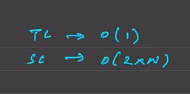
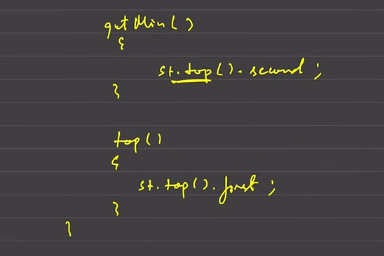
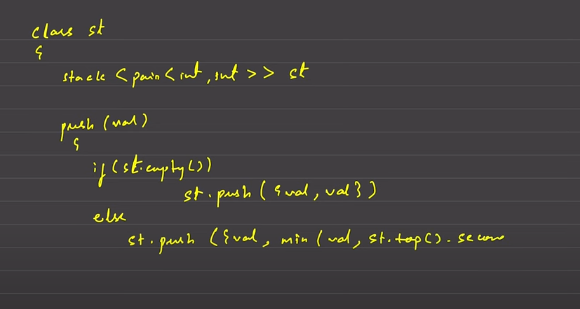


**MIN STACK**

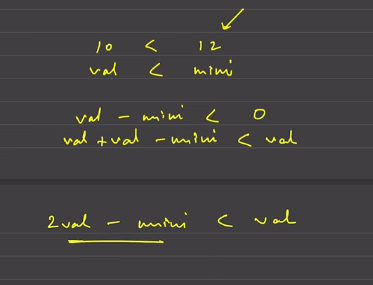
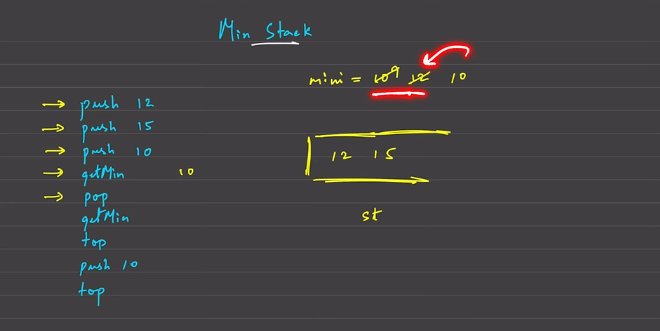
Need to implment the stack operations along with the getMin() function which mst return the minimum among the stack.



To design this from scratch use the linked list.



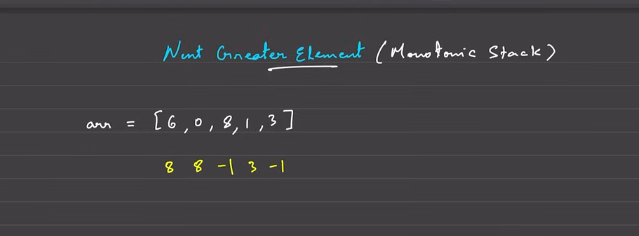
Maintain the stack variable which can manage the minmum number of elements.



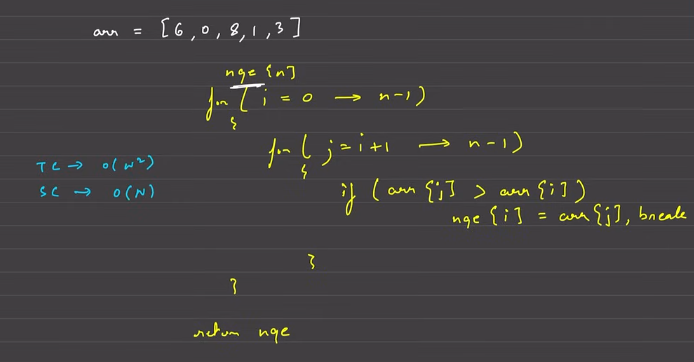
You can use this for revising, but the code will be available in the place where you need it.

**NEXT GREATER ELEMENT**

Monotonic stack; The elements are stored in specific order then it is called as the minotom=nic stack.

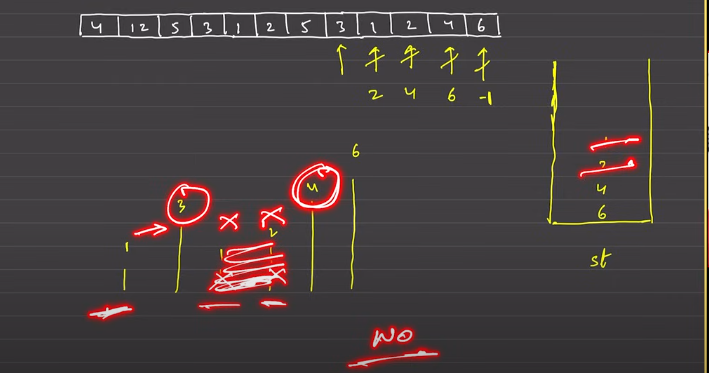
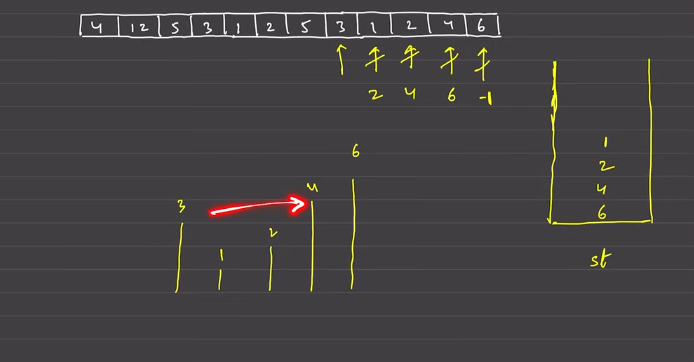
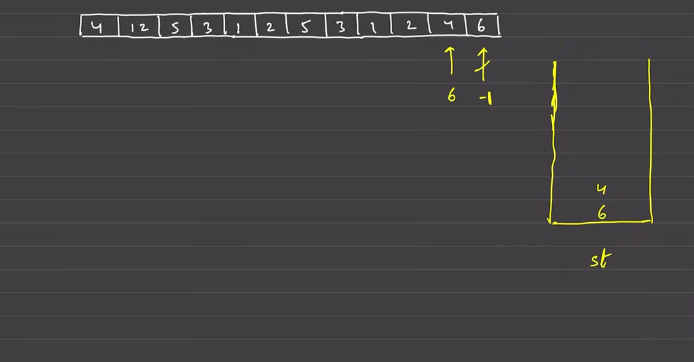


At present index, and the value which is thenext greater elements that is seen in the whole array.   
That is called as the next greater element.

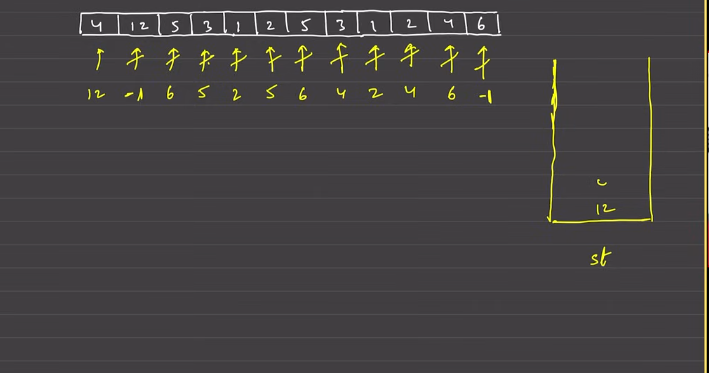
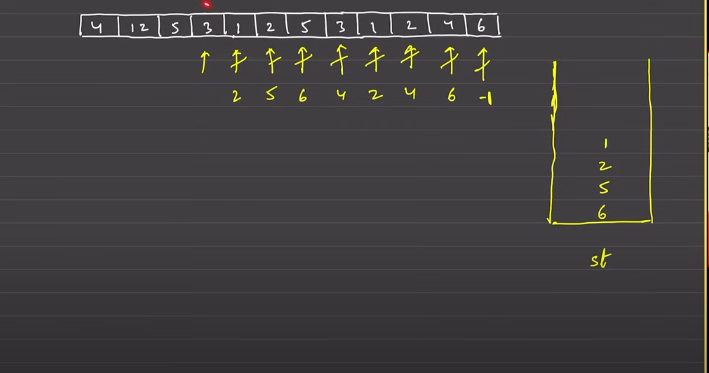


Do not use the same array given as input.

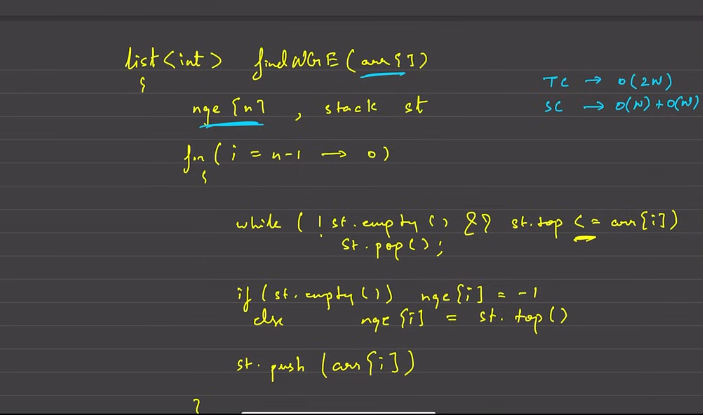
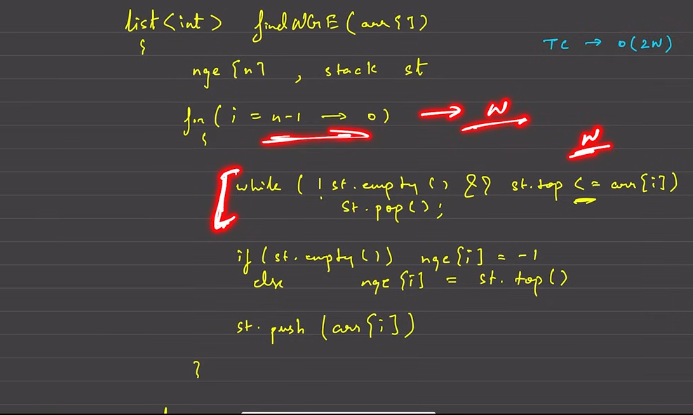
Optimise the above; Use the stack and treverse from the back.



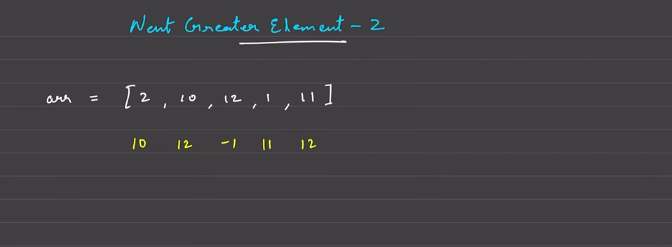
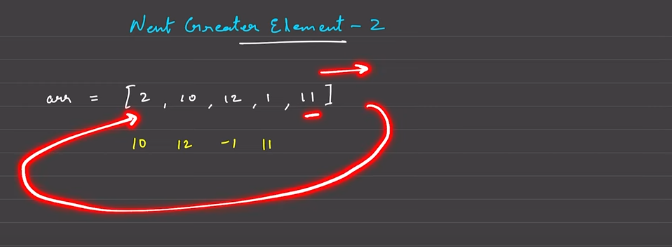
When 3 is there , any body which is there in the left, which are lesser than 3 cannot see anything which are less than 3, so 3 will be the next greater element.



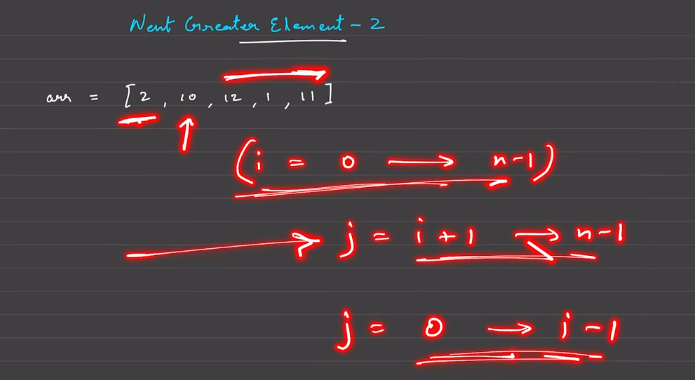
Used the monotonic stack, stored the element in the decreasing order.



Next greater element, seen not only wihtin the array, but the net greater element seen in the circular array.

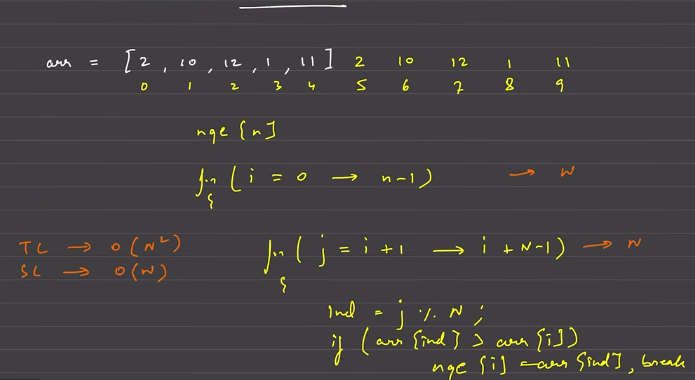
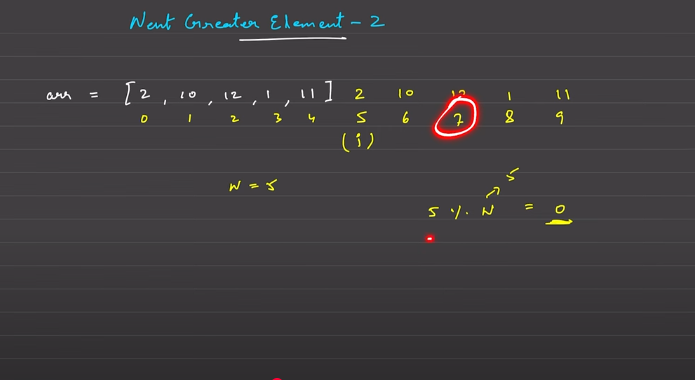


Extreme naive solution.



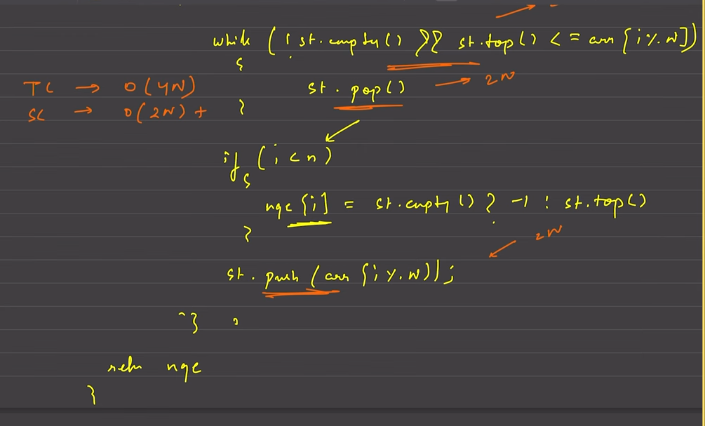
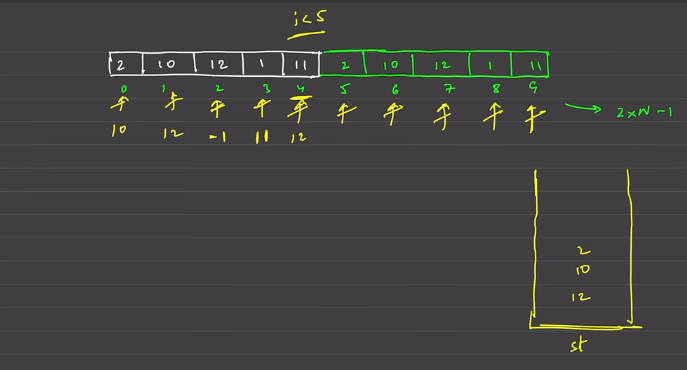
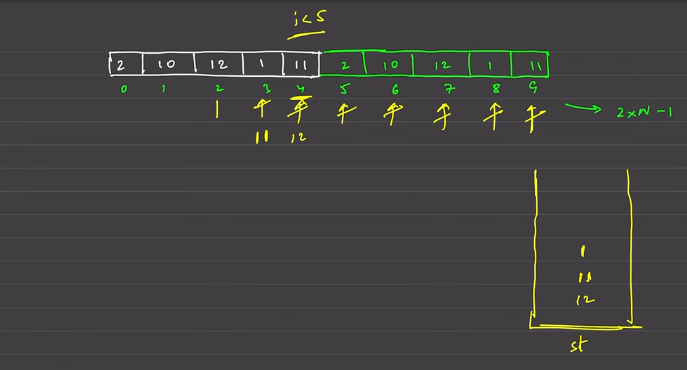
**Optmised solution:**

Circular array:



This is the naive , but clean representatation.

But we need to optkise this till the O(N), so use the monotnoic stack.



**NEAREST SMALER ELEMENTS ON THE LEFT**

